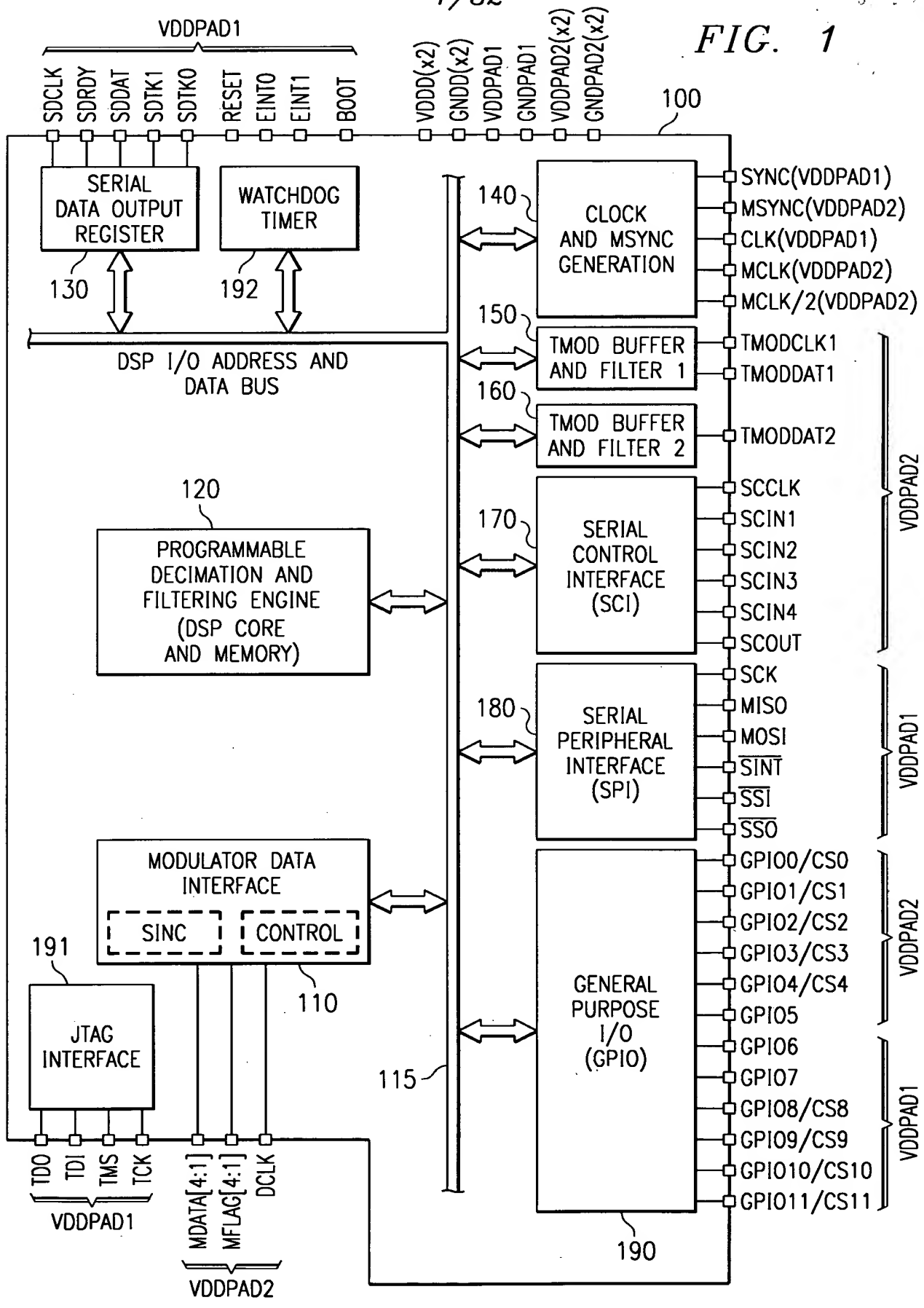
FIG. 1



TDTS90 6thEd86D

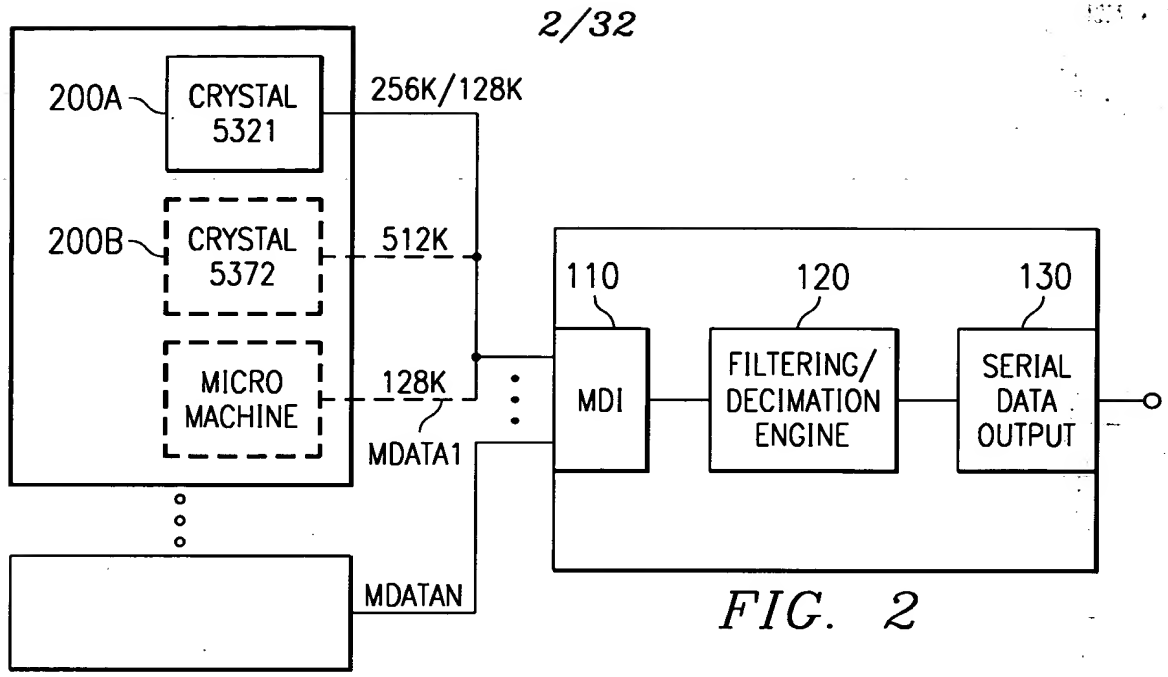


FIG. 2

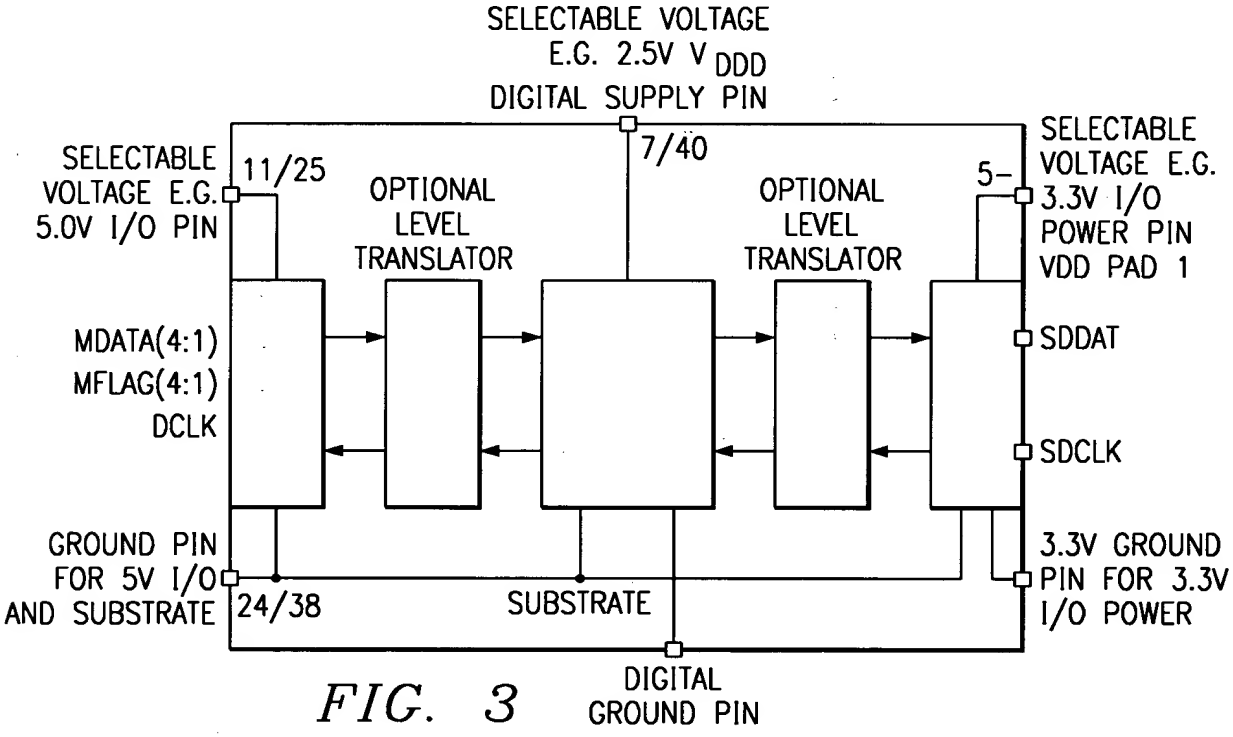


FIG. 3

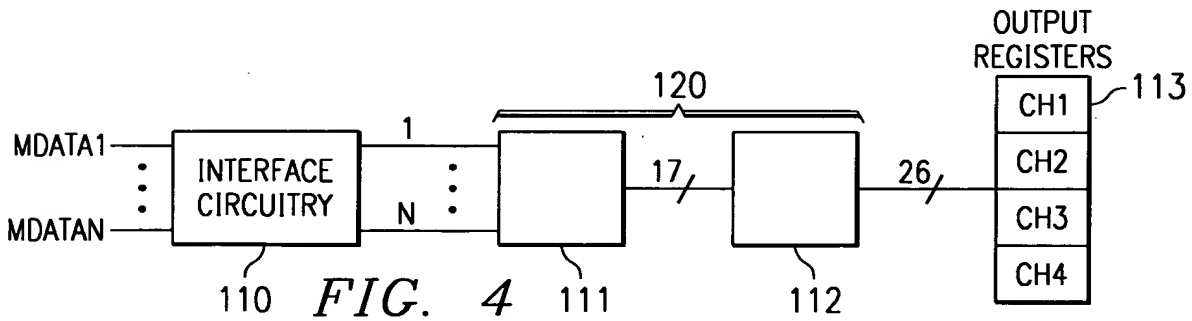


FIG. 4

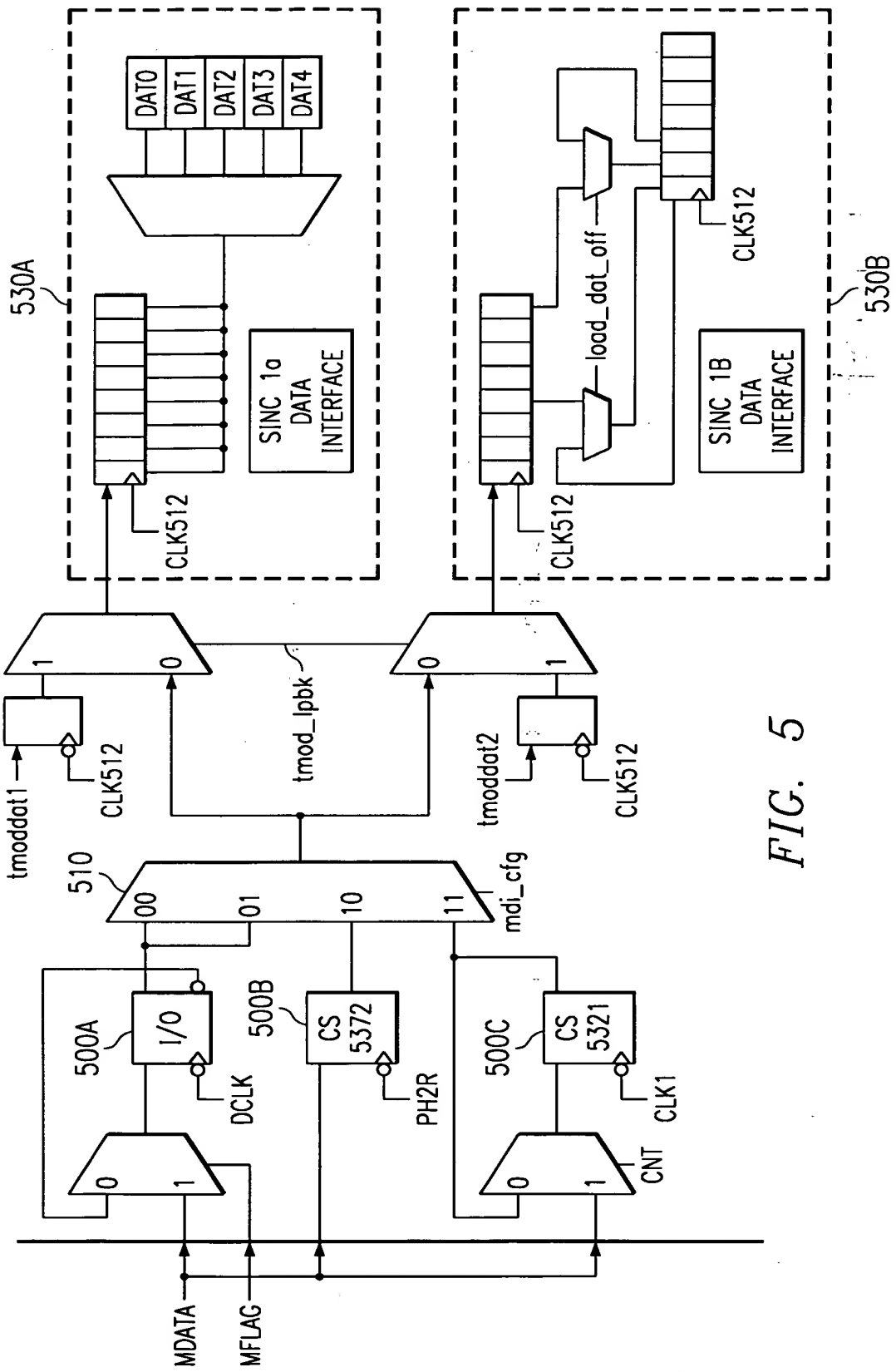
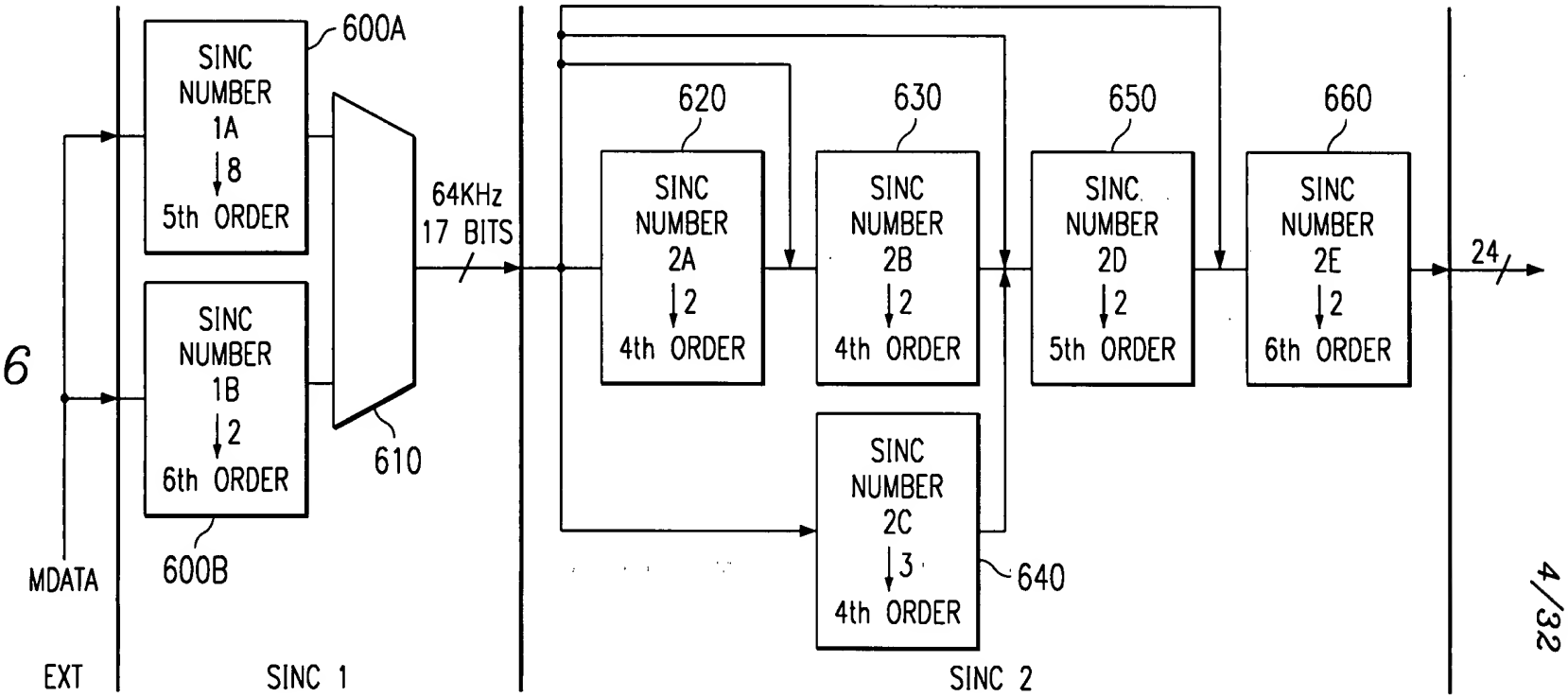

530A
DAT0
DAT1
DAT2
DAT3
DAT4
SINC 1a DATA INTERFACE
CLK512
530B
load_dat_off
CLK512
SINC 1B DATA INTERFACE
CLK512
tmoddat1
CLK512
tmod_lpbk
tmoddat2
CLK512
510
00
01
10
11
mdi_cfg
500A
I/O
DCLK
500B
CS 5372
PH2R
500C
CS 5321
CLK1
CNT
MDATA
MFLAG

FIG. 5

FIG. 6



4/32

FIG. 7

- FIFTH ORDER DECIMATE BY 8:

$$H(z)=\left(\frac{1-z^{-8}}{1-z^{-1}}\right)^{5}$$

- 36 TAP FIR FILTER. HALF OF THE (SYMMETRIC) COEFFICIENTS

$h_0=1$	$h_1=5$	$h_2=15$	$h_3=35$	$h_4=70$	$h_5=126$	$h_6=210$	$h_7=330$	$h_8=490$
$h_9=690$	$h_{10}=926$	$h_{11}=1190$	$h_{12}=1470$	$h_{13}=1750$	$h_{14}=2010$	$h_{15}=2226$	$h_{16}=2380$	$h_{17}=2460$

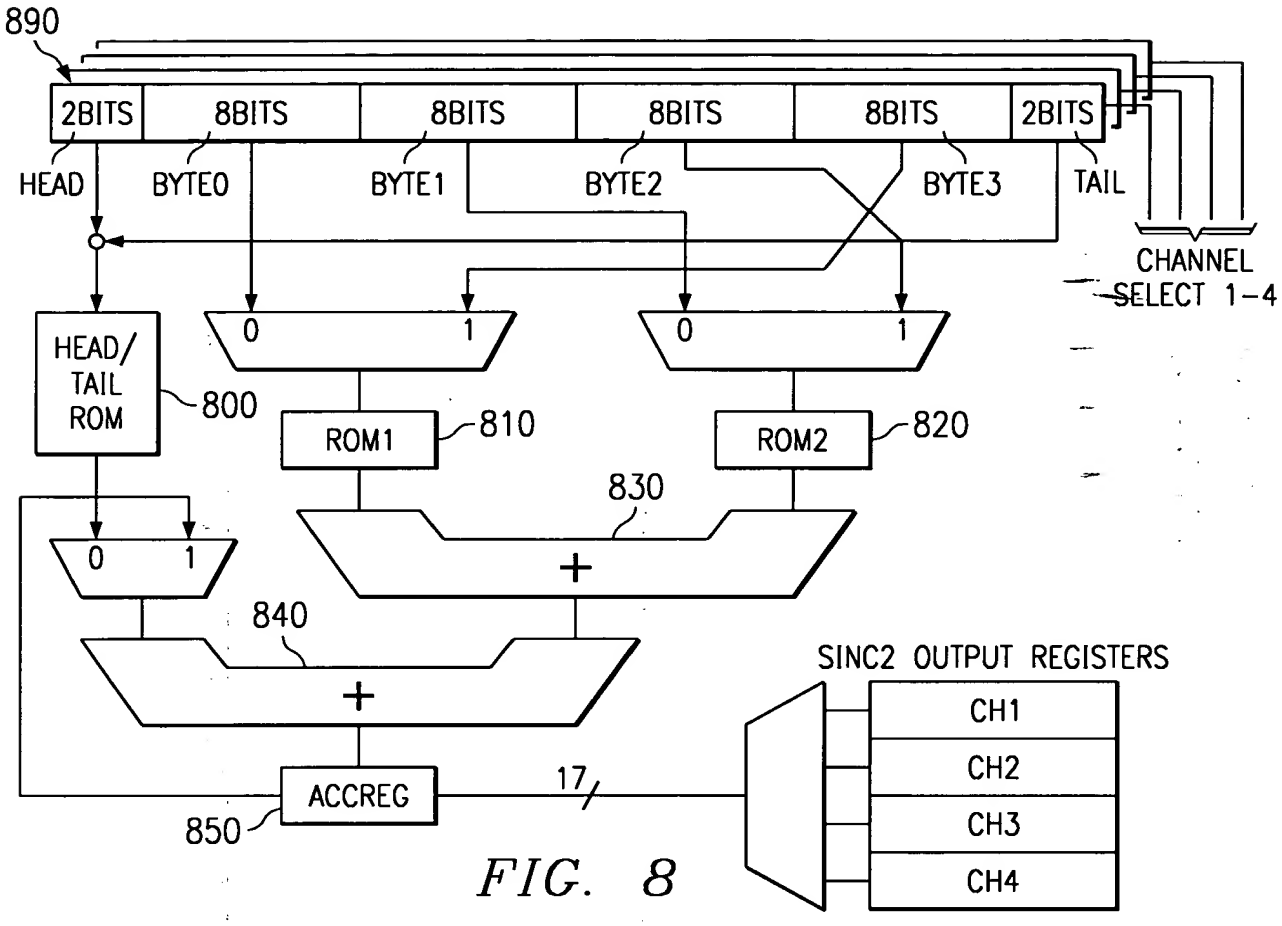


FIG. 8

$$H(z)=\left(\frac{1-z^{-2}}{1-z^{-1}}\right)^{6}$$

IMPULSE RESPONSE:

$$y[n]=x[n]+6\cdot x[n-1]+15\cdot x[n-2]+20\cdot x[n-3]+15\cdot x[n-4]+6\cdot x[n-5]+x[n-6]$$

FIG. 9

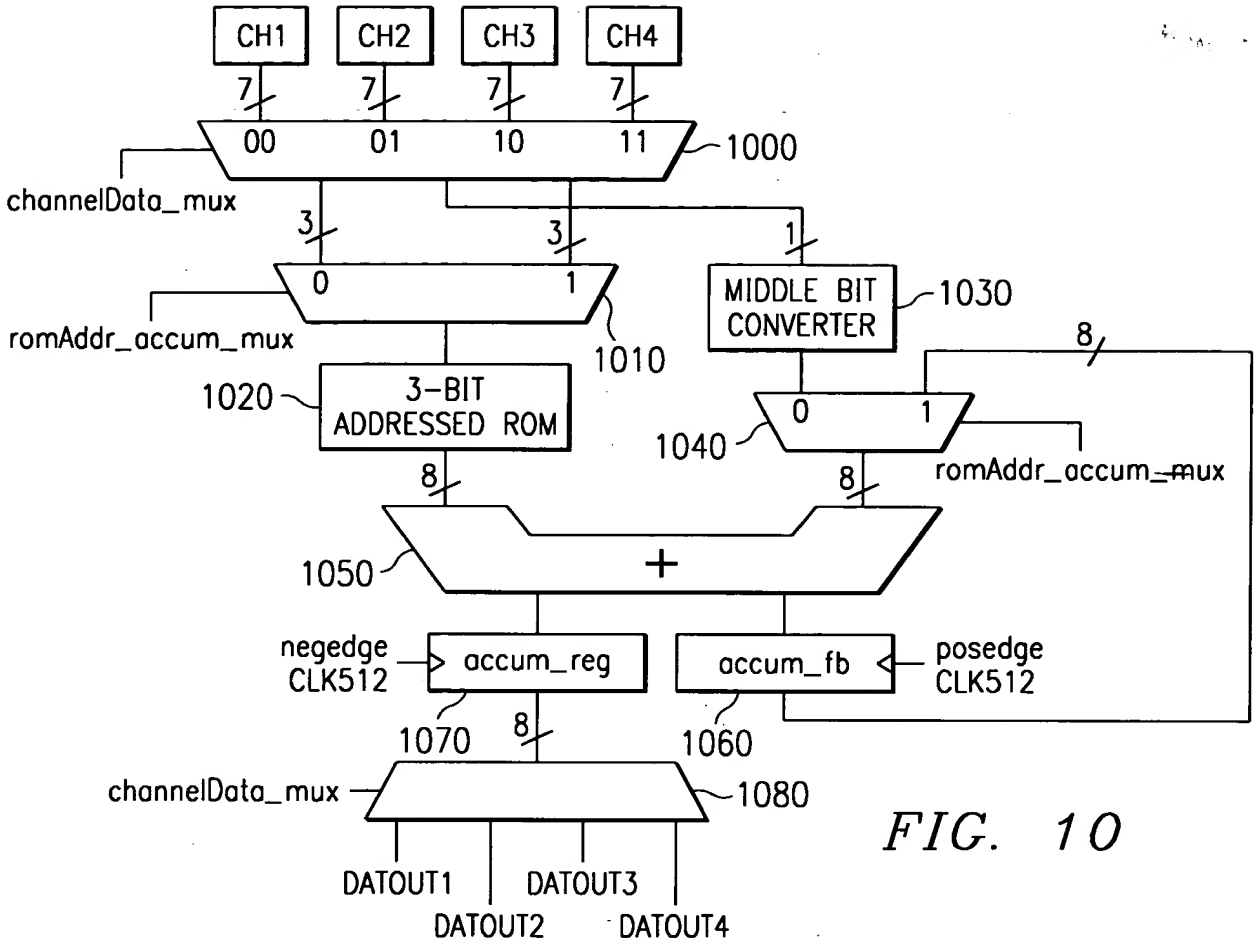


FIG. 10

FILTER NAME	SYSTEM FUNCTION	IMPULSE RESPONSE (FILTER COEFFICIENTS)
Sinc2(a) Sinc2(b)	$H(z)=\left(\frac{1-z^{-2}}{1-z^{-1}}\right)^4$	h[n]=[1 4 6 4 1]
Sinc2(c)	$H(z)=\left(\frac{1-z^{-3}}{1-z^{-1}}\right)^4$	h[n]=[1 4 10 16 19 16 10 4 1]
Sinc2(d)	$H(z)=\left(\frac{1-z^{-2}}{1-z^{-1}}\right)^5$	h[n]=[1 5 10 10 5 1]
Sinc2(e)	$H(z)=\left(\frac{1-z^{-2}}{1-z^{-1}}\right)^6$	h[n]=[1 6 15 20 15 6 1]

FIG. 11

Fotofo: 6tE0860

Sinc2(a) and Sinc2(b):

$$y[n]=x[n]+4x[n-1]+6x[n-2]+4x[n-3]+x[n-4]$$
$$=x[n]+4x[n-1]+4x[n-2]+2x[n-2]+4x[n-3]+x[n-4]$$

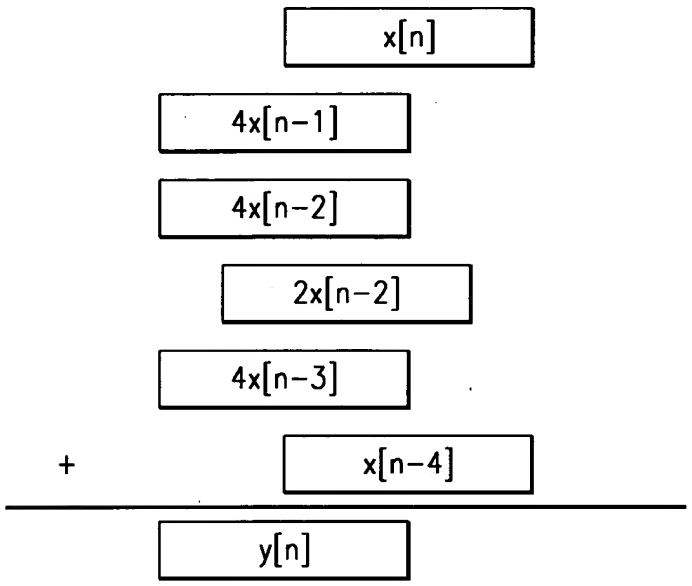


FIG. 12

Sinc2(a) and Sinc2(b):

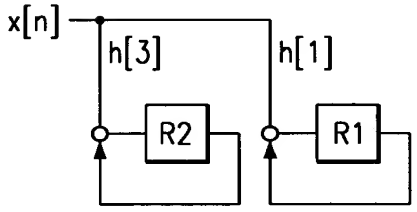


FIG. 14A

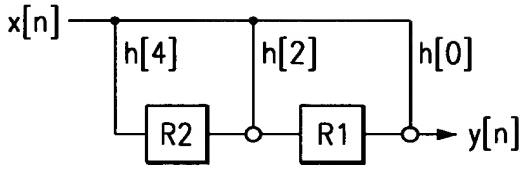


FIG. 14B

Sinc2(d):

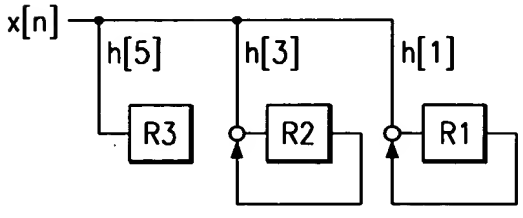


FIG. 15A

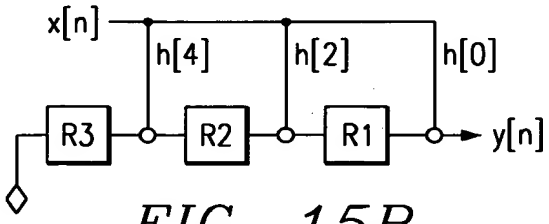


FIG. 15B

Fotofab-6343860

FIG. 13A

Sinc2(c):

$$y[n] = x[n]+4x[n-1]+10x[n-2]+16x[n-3]+19x[n-4]+16x[n-5]+10x[n-6]+4x[n-7]+x[n-8]$$
$$= x[n]+4x[n-1]+\boxed{8x[n-2]+2x[n-2]}+16x[n-3]+\boxed{16x[n-4]+2x[n-4]+x[n-4]}$$
$$+16x[n-5]+\boxed{8x[n-6]+2x[n-6]}+4x[n-7]+x[n-8]$$

FIG. 13B

Sinc2(d):

$$y[n] = x[n]+5x[n-1]+10x[n-2]+10x[n-3]+5x[n-4]+x[n-5]$$
$$= x[n]+\boxed{4x[n-1]+x[n-1]}+\boxed{8x[n-2]+2x[n-2]}+\boxed{8x[n-3]+2x[n-3]}+\boxed{4x[n-4]+x[n-4]}+x[n-5]$$

FIG. 13C

Sinc2(e):

$$y[n] = x[n]+6x[n-1]+15x[n-2]+20x[n-3]+15x[n-4]+6x[n-5]+x[n-6]$$
$$= x[n]+\boxed{4x[n-1]+2x[n-1]}+\boxed{16x[n-2]-x[n-2]}+\boxed{16x[n-3]+4x[n-3]}$$
$$+\boxed{16x[n-4]-x[n-4]}+\boxed{4x[n-5]+2x[n-5]}+x[n-6]$$

Sinc2(c):

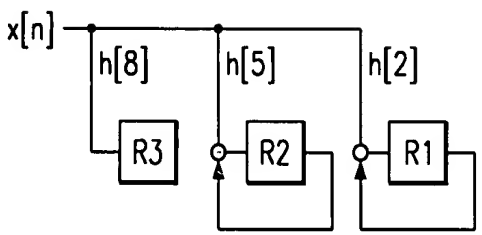


FIG. 16A

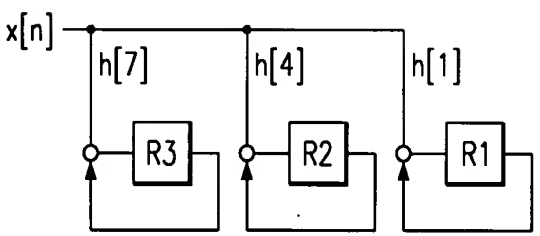


FIG. 16B

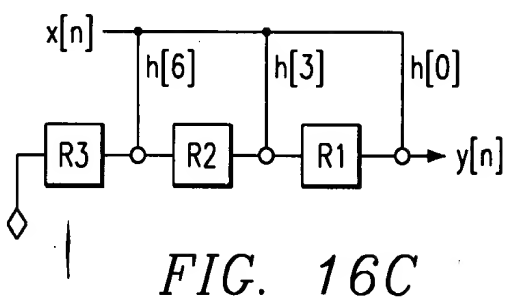


FIG. 16C

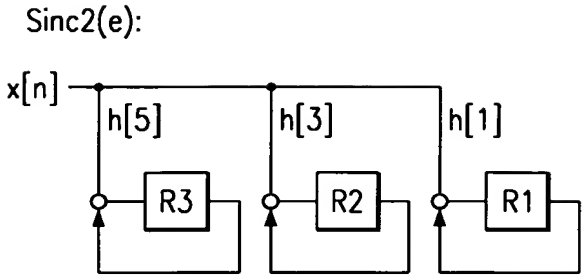


FIG. 17A

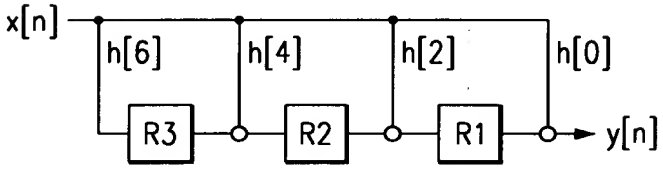


FIG. 17B

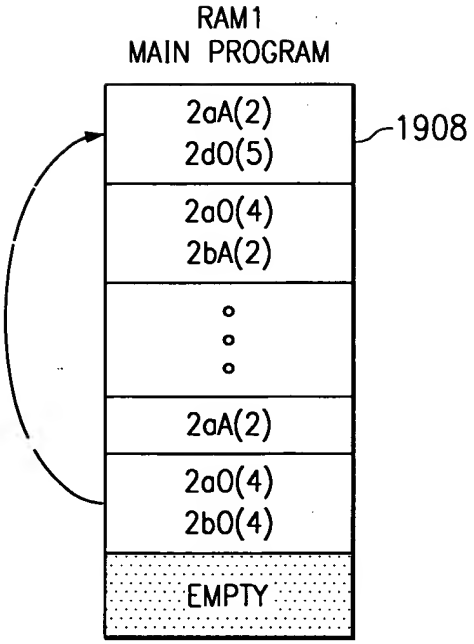


FIG. 19A

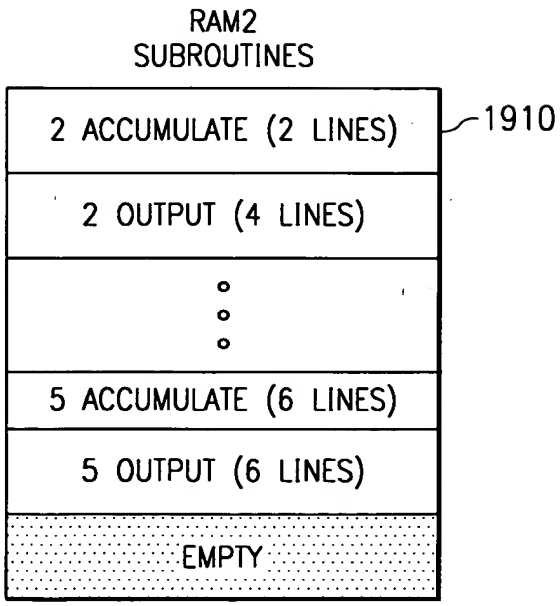


FIG. 19B

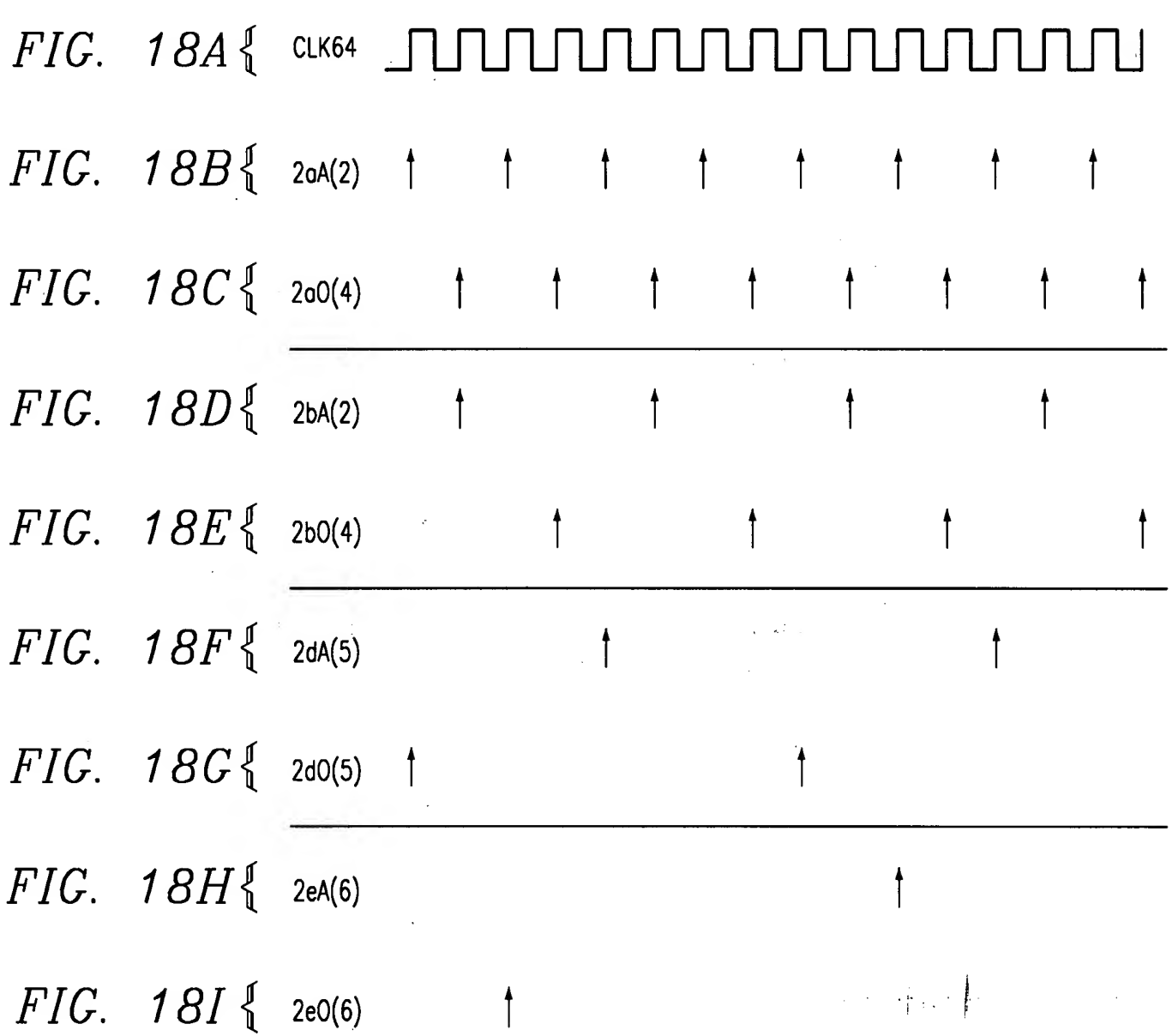


FIG. 18A
CLK64
FIG. 18B
2aA(2)
FIG. 18C
2aO(4)
FIG. 18D
2bA(2)
FIG. 18E
2bO(4)
FIG. 18F
2dA(5)
FIG. 18G
2dO(5)
FIG. 18H
2eA(6)
FIG. 18I
2eO(6)

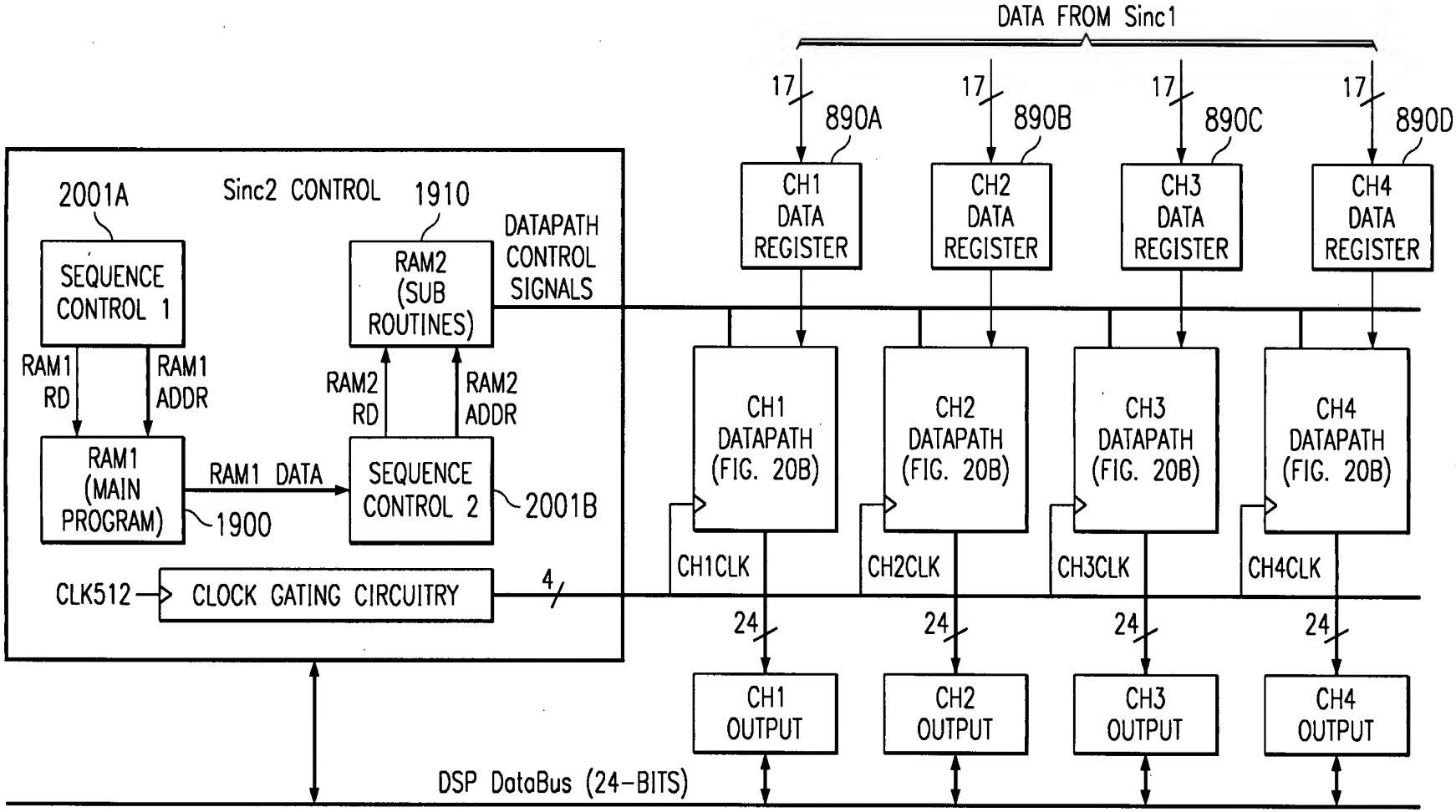

DATA FROM Sinc1
17
890A
890B
890C
890D
CH1 DATA REGISTER
CH2 DATA REGISTER
CH3 DATA REGISTER
CH4 DATA REGISTER
2001A
Sinc2 CONTROL
1910
SEQUENCE CONTROL 1
RAM2 (SUB ROUTINES)
DATAPATH CONTROL SIGNALS
RAM1 RD
RAM1 ADDR
RAM2 RD
RAM2 ADDR
RAM1 (MAIN PROGRAM)
1900
RAM1 DATA
SEQUENCE CONTROL 2
2001B
CH1 DATAPATH (FIG. 20B)
CH2 DATAPATH (FIG. 20B)
CH3 DATAPATH (FIG. 20B)
CH4 DATAPATH (FIG. 20B)
CLK512
CLOCK GATING CIRCUITRY
4
CH1CLK
CH2CLK
CH3CLK
CH4CLK
24
CH1 OUTPUT
CH2 OUTPUT
CH3 OUTPUT
CH4 OUTPUT
DSP DataBus (24-BITS)

FIG. 20A

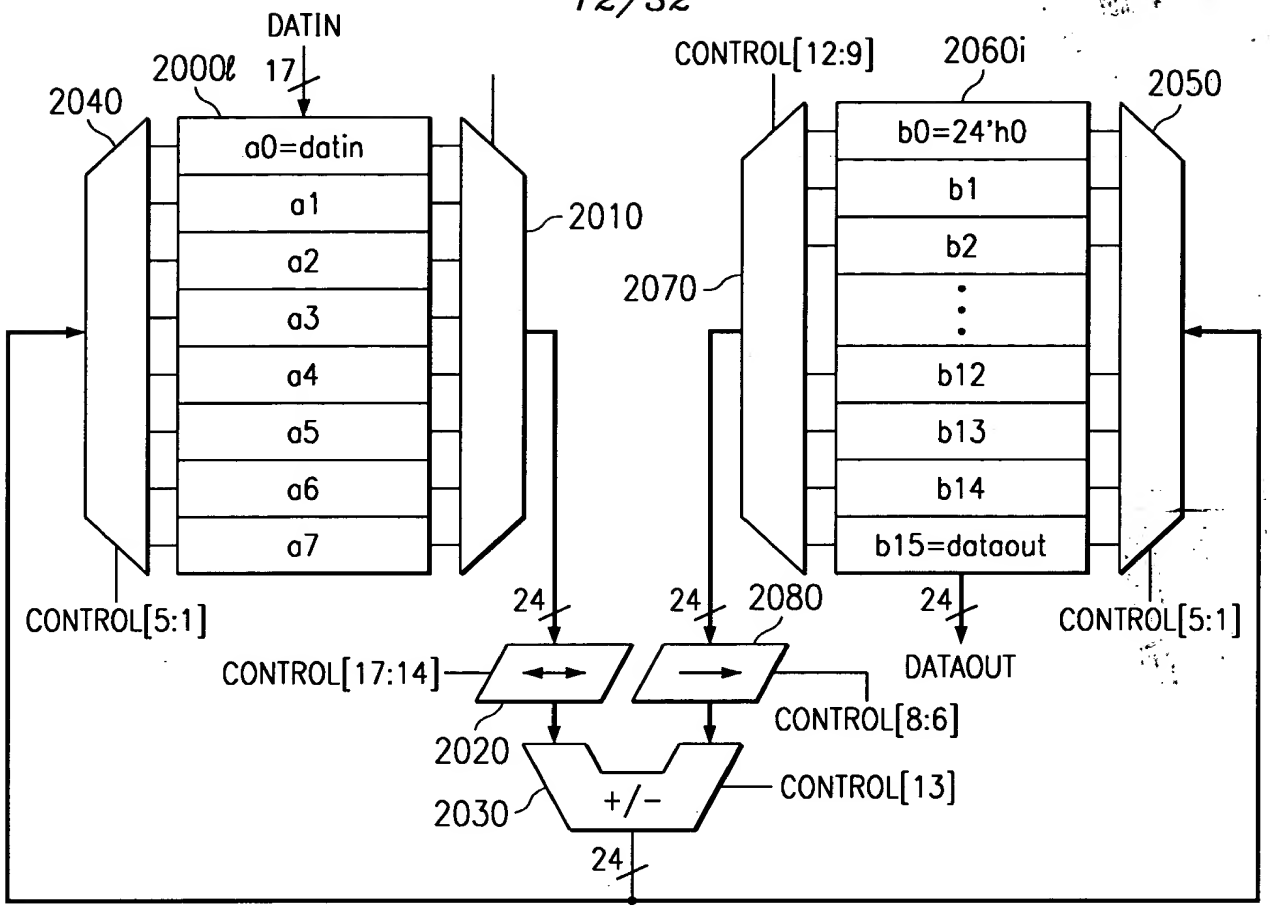


FIG. 20B

PROGRAMMING PROCEDURE:

1. SELECT DECIMATION RATE.

2. SELECT REQUIRED MINI-SINCS AND ASSOCIATAED ACCUMULATE AND OUTPUT SUBROUTINES.

3. SEPARATE COEFFICIENTS INTO FORM SUITABLE FOR SHIFT-ADD OPERATIONS.

4. CHECK FOR OVERFLOW AFTER EACH ADDITION IN THE FILTER.

5. PERFORM NECESSARY TRUNCATION TO 24 BITS AND SCALING OF SUBSEQUENT COEFFICIENTS IN MINI-SINCS.

6. TIME MULTIPLEX ACCUMULATE AND OUTPUT SUBROUTINES SO THAT A MAXIMUM OF 8 ADDITIONS/SUBTRACTIONS ARE PERFORMED FOR EACH INPUT FROM SINC1.

7. CREATE CODE FOR RAM2 (ACCUMULATE AND OUTPUT SUBROUTINES) IN THE FORM: [Coeff 1] [Src 1] [Src 2] [Dest] [Coeff2] [Done Subroutine]

8. CREATE CODE FOR RAM1 (MAIN CONTROL CODE) [Line #] [Wait for new data] [Done program]

FIG. 21

TDTF90: 5HE0860

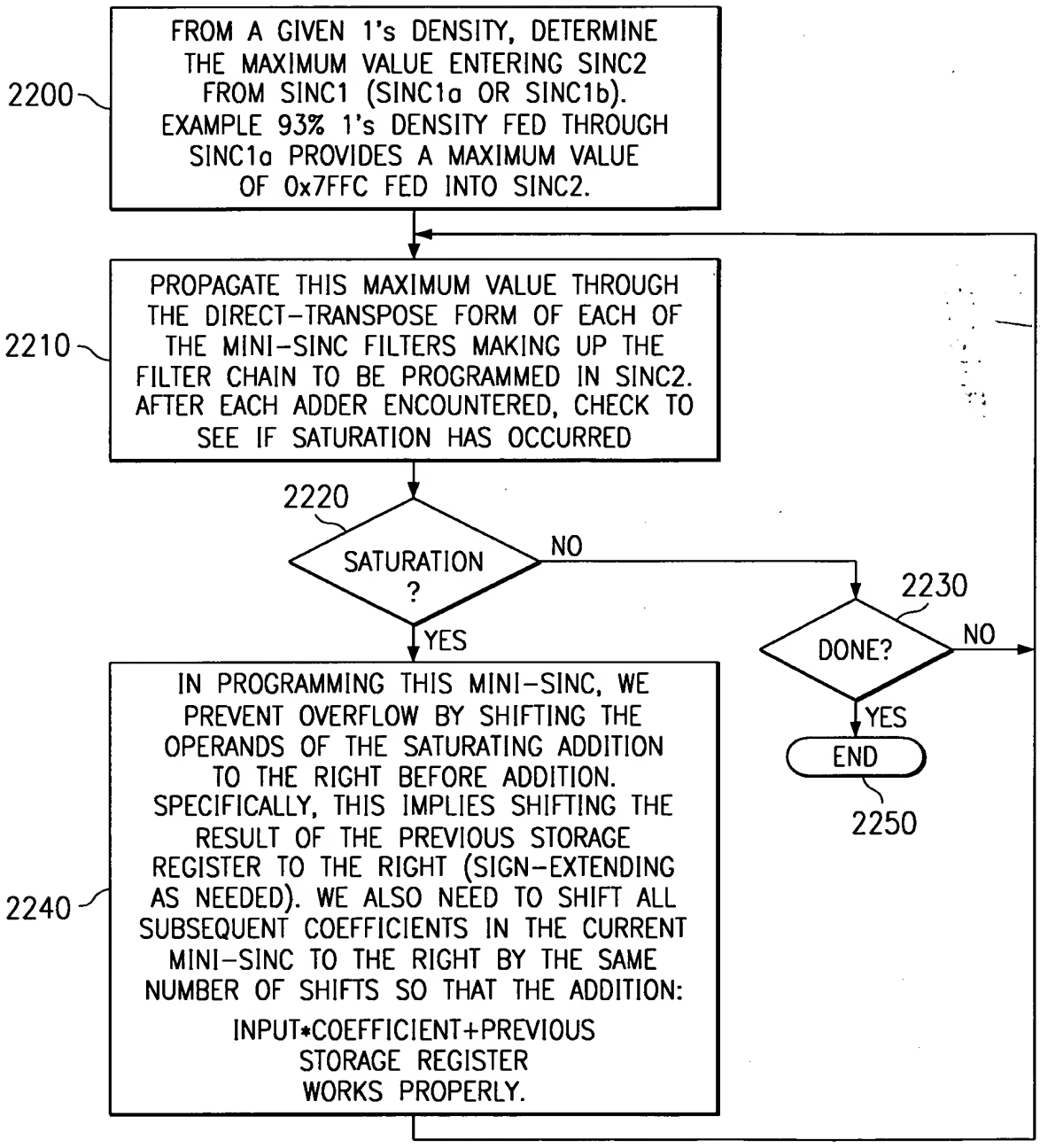


FIG. 22

TOHTSO:6H3E0860

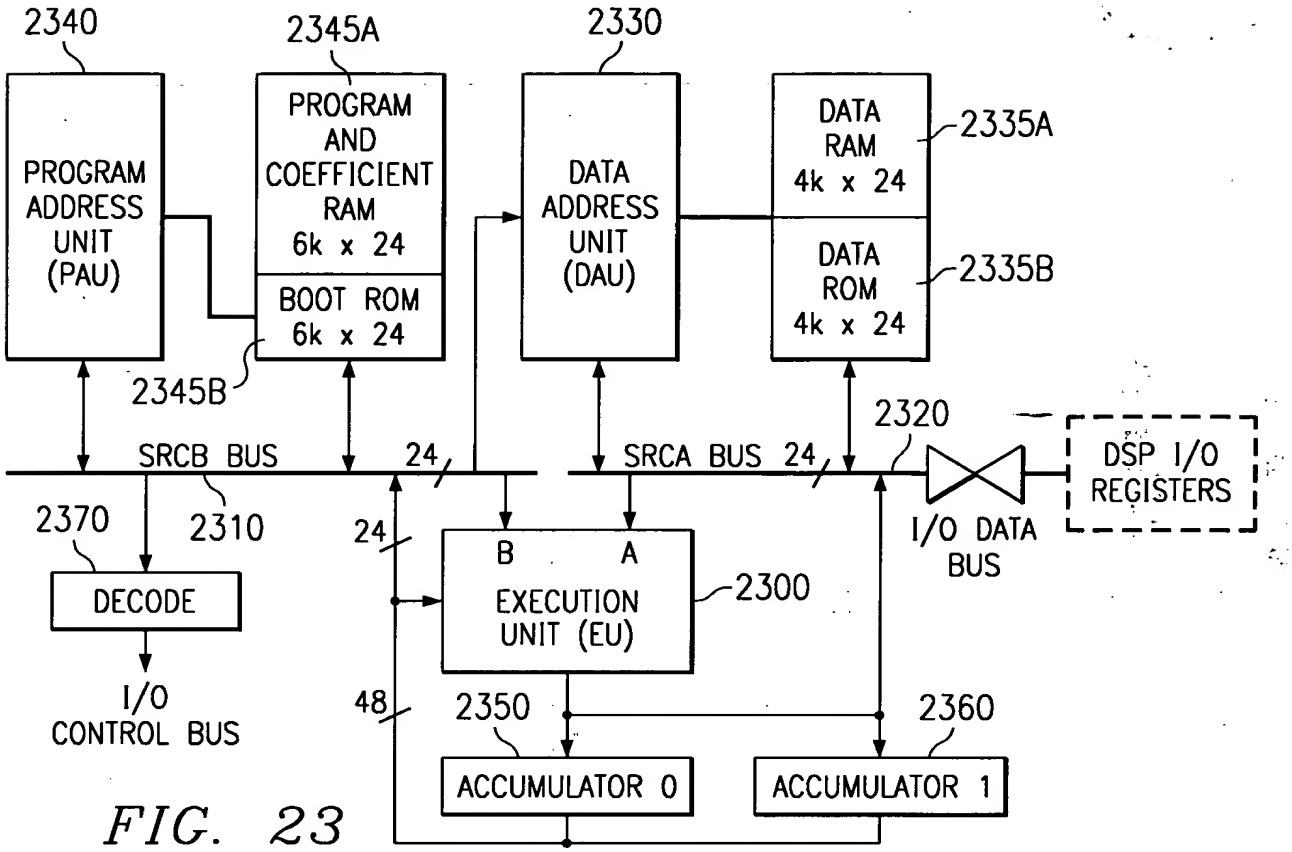


FIG. 23

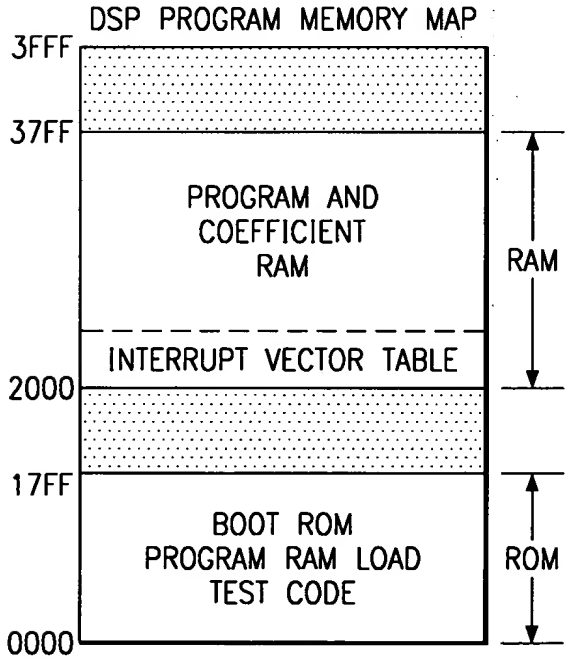


FIG. 24A

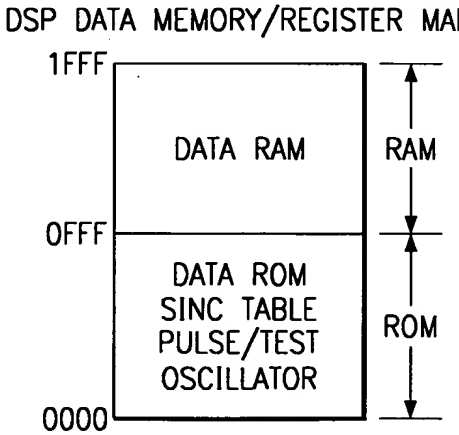


FIG. 24B

TOHTSO:6+E0E80

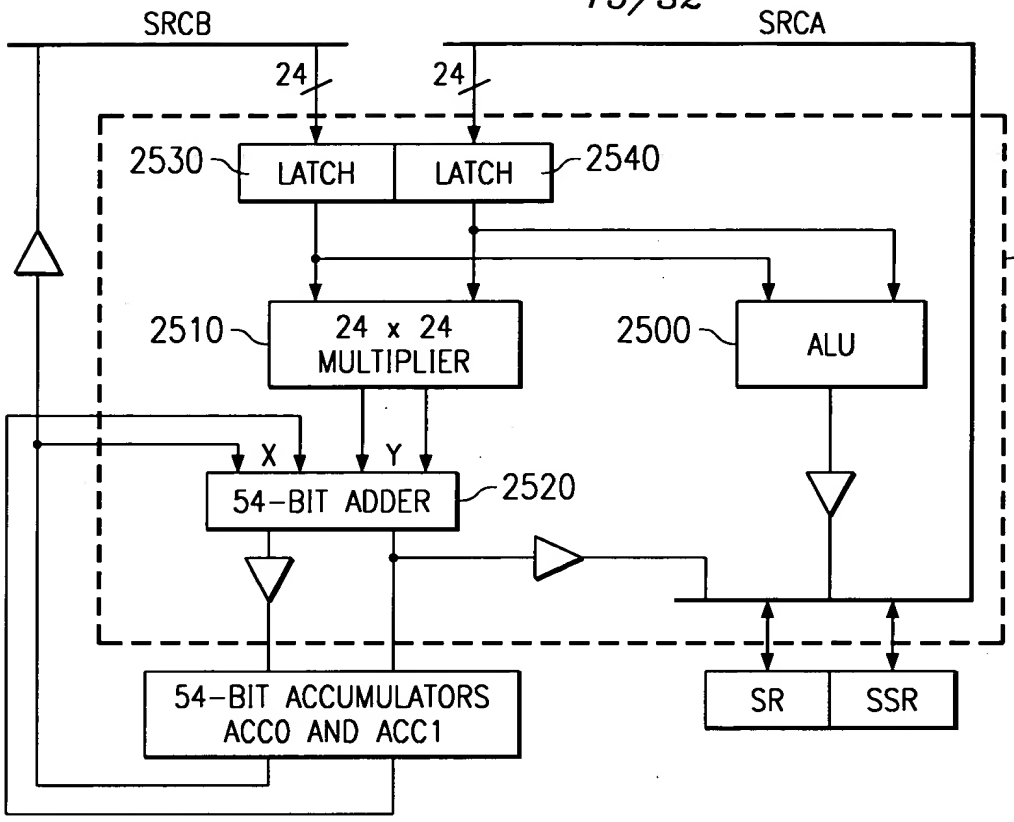

SRCB
SRCA
24
24
2530
LATCH
LATCH
2540
2300
2510
24 x 24
MULTIPLIER
2500
ALU
X
Y
54-BIT ADDER
2520
54-BIT ACCUMULATORS
ACC0 AND ACC1
SR
SSR

FIG. 25

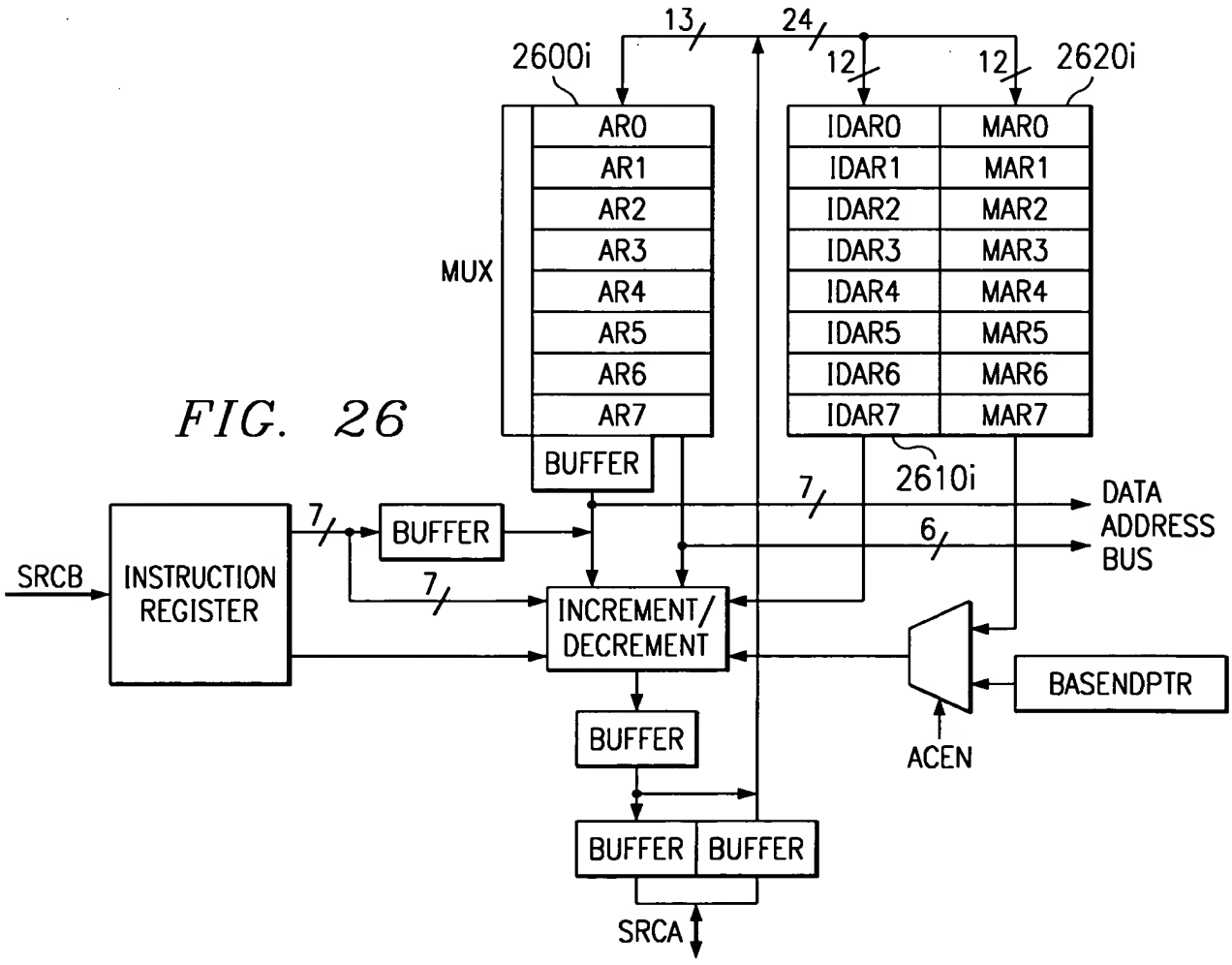

13
24
2600i
12
12
2620i
AR0
AR1
AR2
AR3
AR4
AR5
AR6
AR7
MUX
IDAR0
IDAR1
IDAR2
IDAR3
IDAR4
IDAR5
IDAR6
IDAR7
MAR0
MAR1
MAR2
MAR3
MAR4
MAR5
MAR6
MAR7
BUFFER
2610i
7
DATA
ADDRESS
BUS
6
SRCB
INSTRUCTION
REGISTER
7
BUFFER
7
INCREMENT/
DECREMENT
BASENDPTR
ACEN
BUFFER
BUFFER
BUFFER
SRCA

FIG. 26

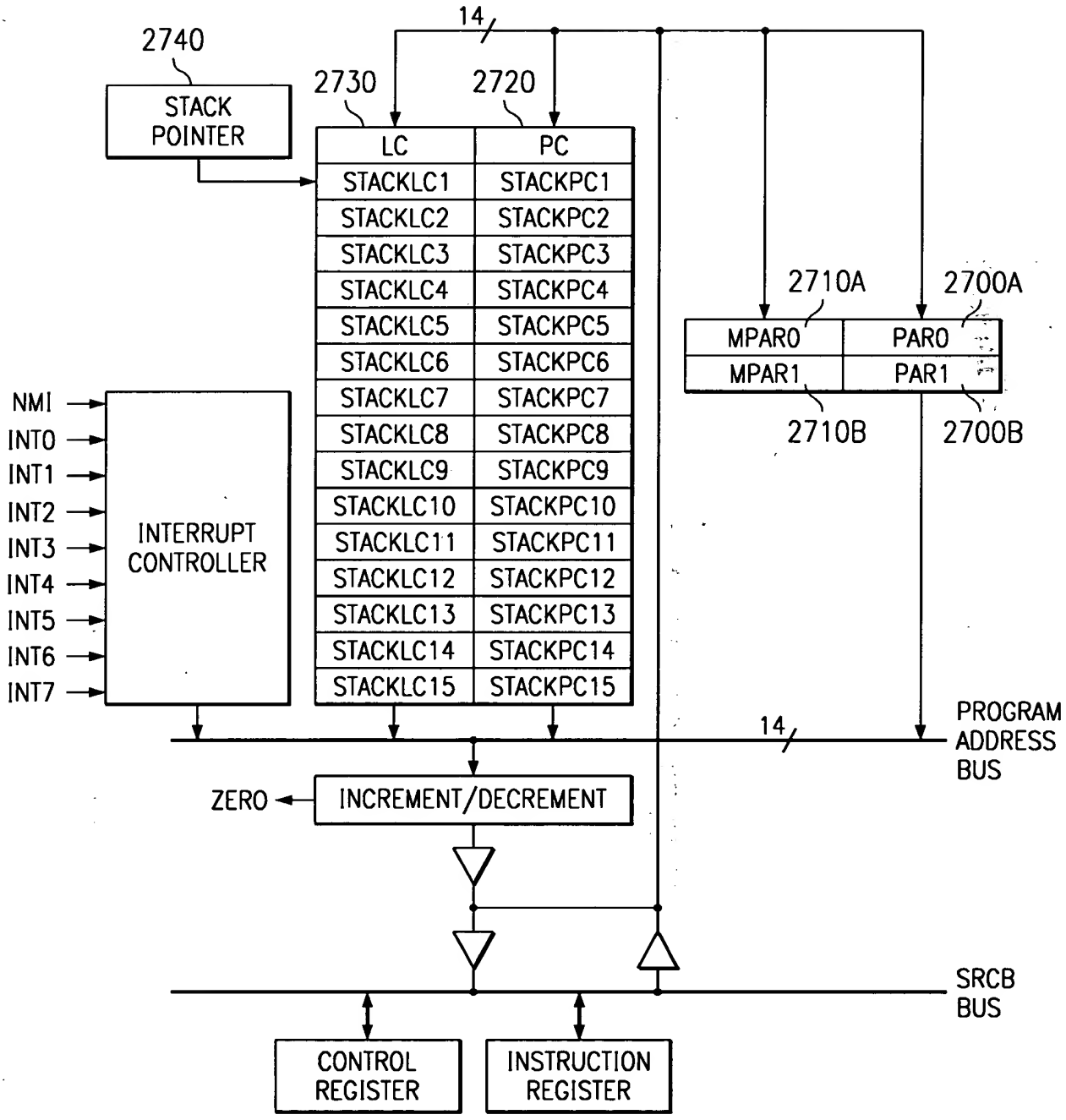


FIG. 27

T-04T90-6+E3+0860

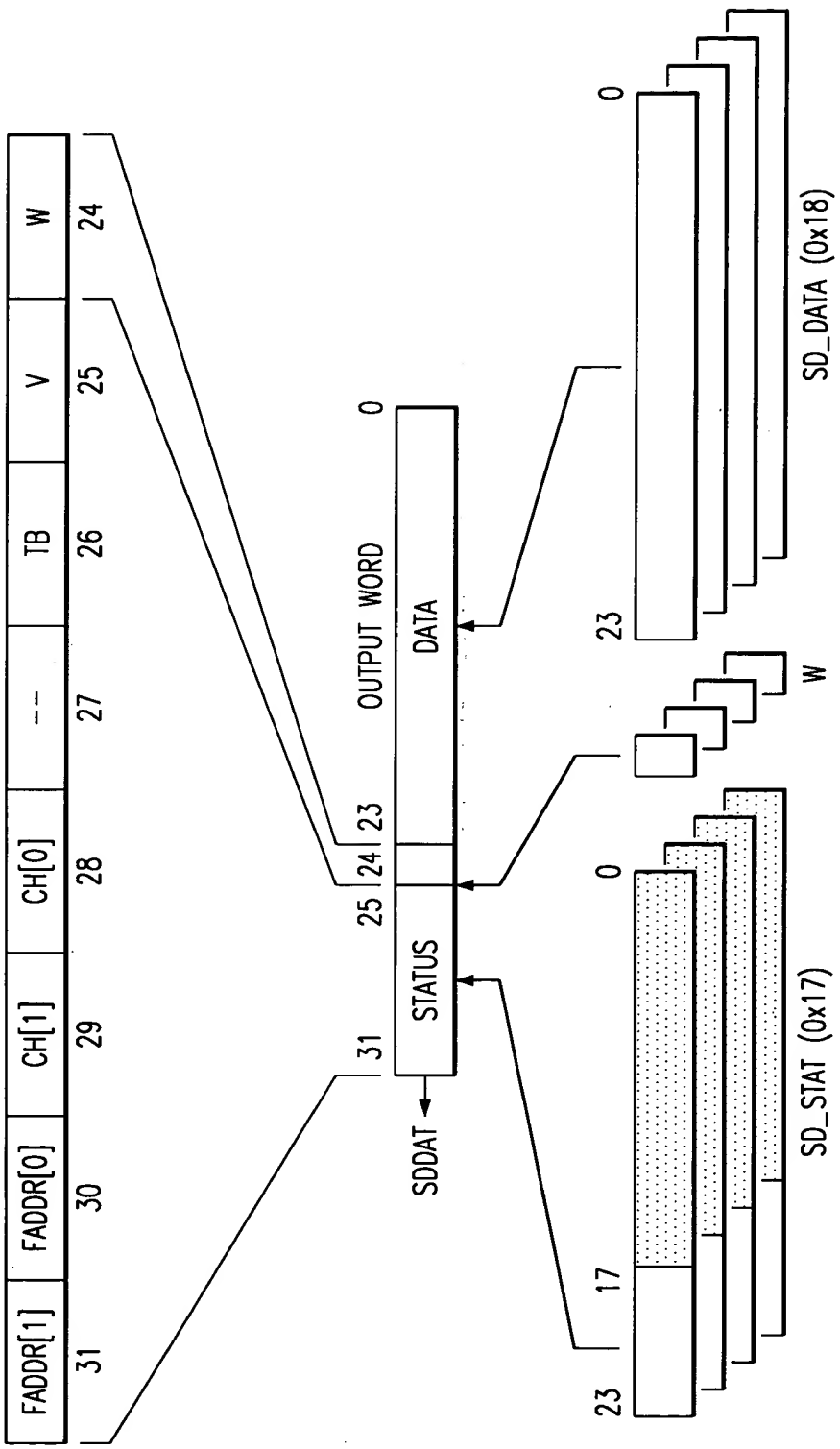

FADDR[1]
FADDR[0]
CH[1]
CH[0]
--
TB
V
W
31
30
29
28
27
26
25
24
23
0
OUTPUT WORD
DATA
STATUS
SDDAT
SD_DATA (0x18)
SD_STAT (0x17)
W

FIG. 28

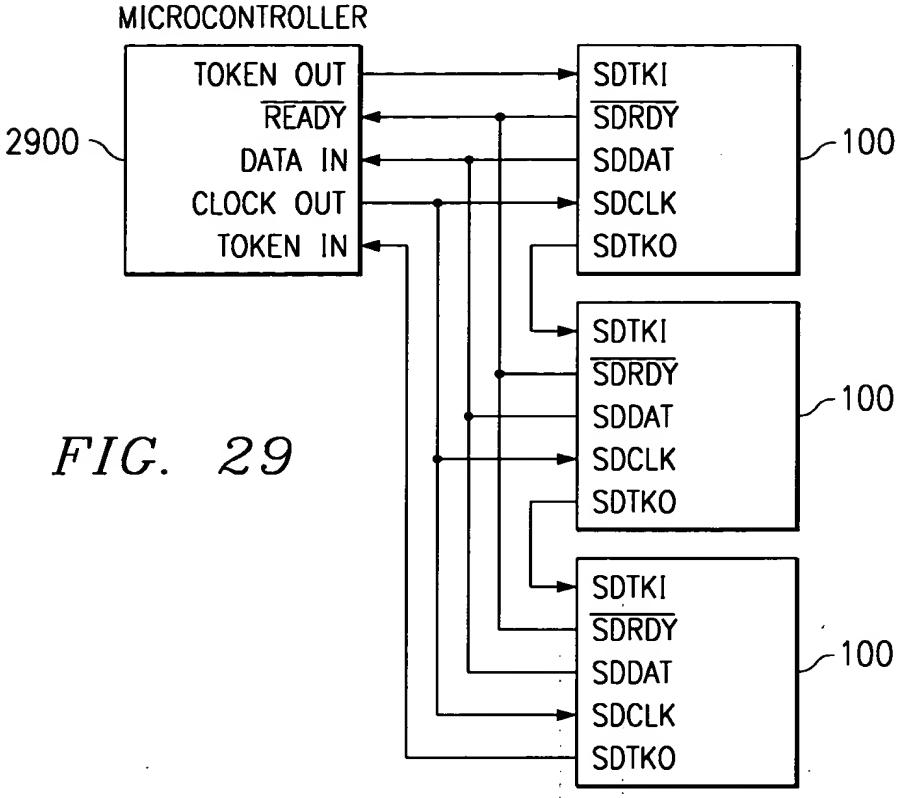


FIG. 29

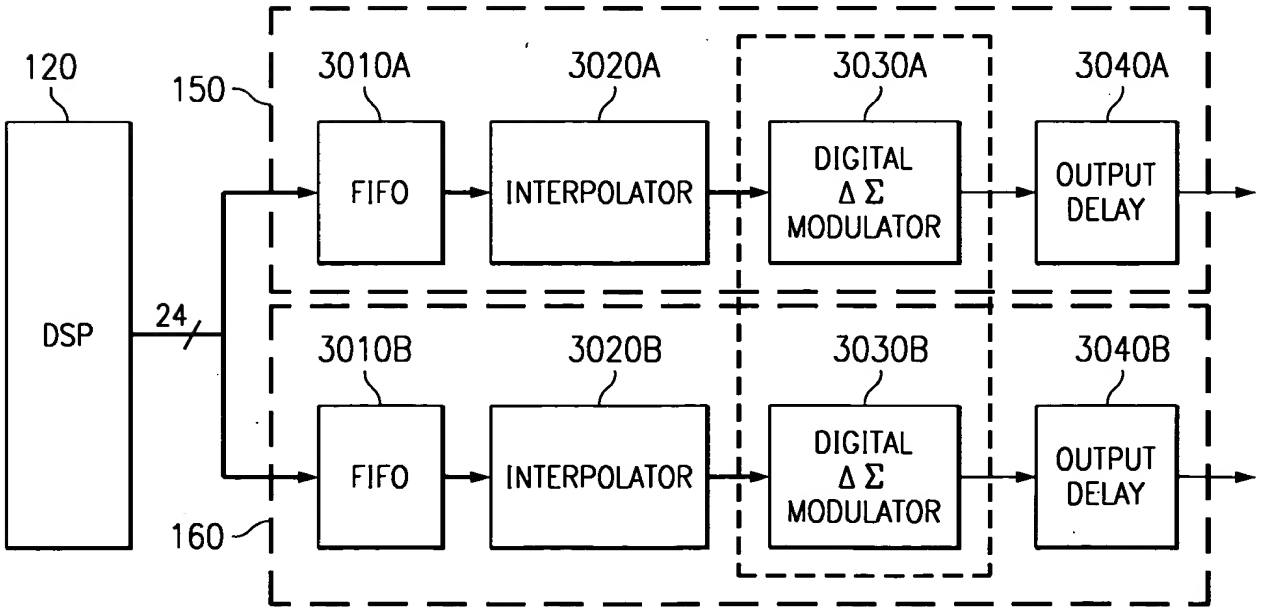


FIG. 30A

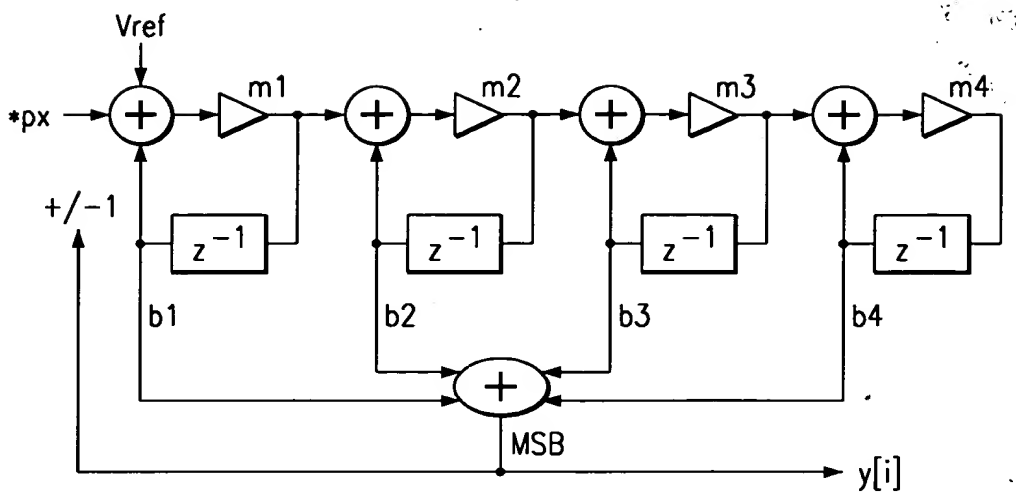


FIG. 30B

FIG. 30C−1 WIRE

FIG. 30C−2 24 WIRES

FIG. 30C−3 REGISTER

FIG. 30C−4 MULTIPLEXER

FIG. 30C−5 TRISTATE BUFFER

FIG. 30C−6 INVERTER

FIG. 30C−7 EXCLUSIVE OR GATE

FIG. 30C−8 ADDER

FIG. 30C−9 MULTIPLIER

FIG. 30C−10 RIGHT SHIFTER

Doc#: 5,483,256

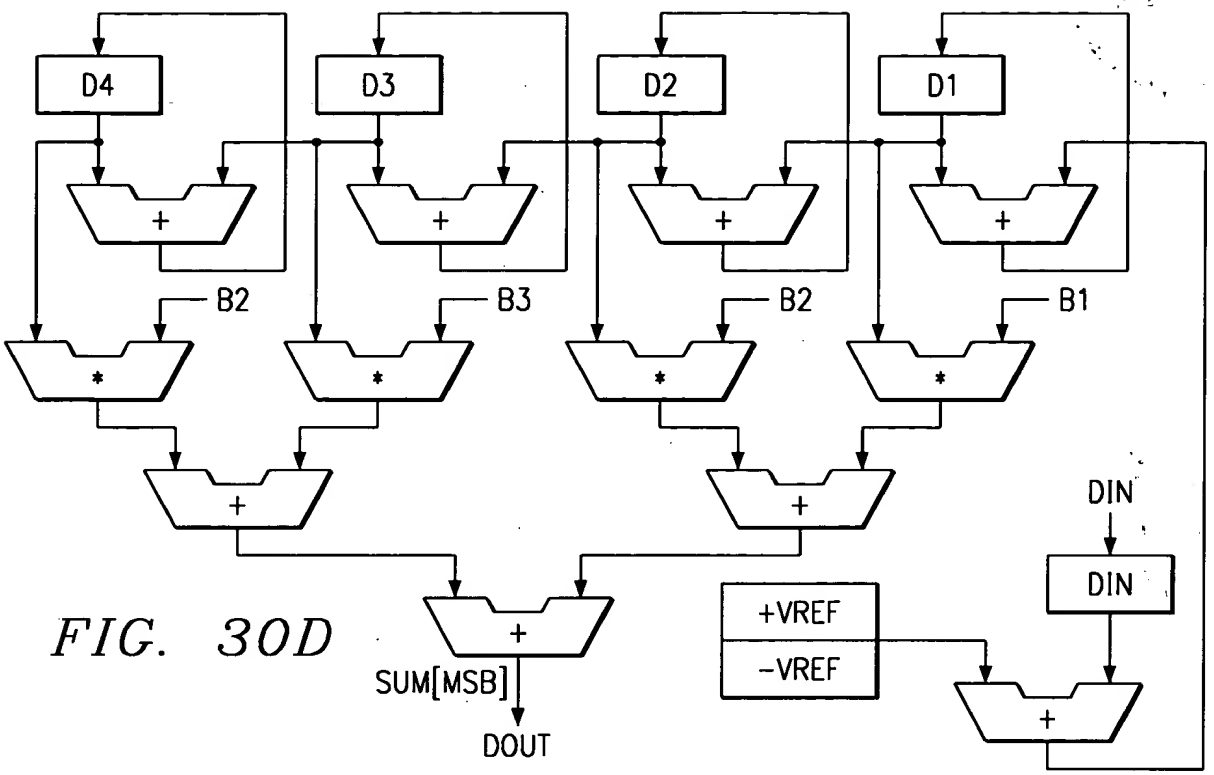


FIG. 30D

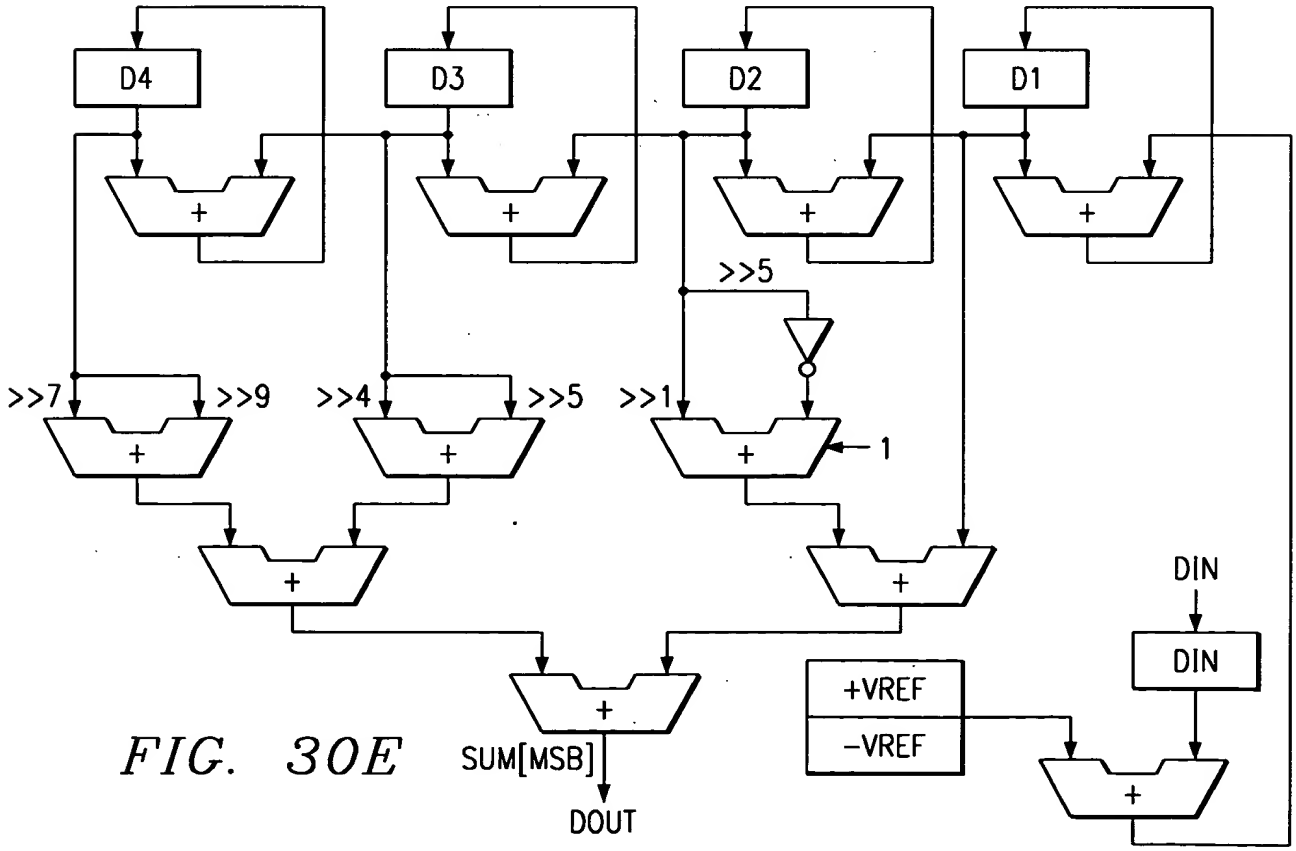


FIG. 30E

T04T50-054E0850

FIG. 30F-1

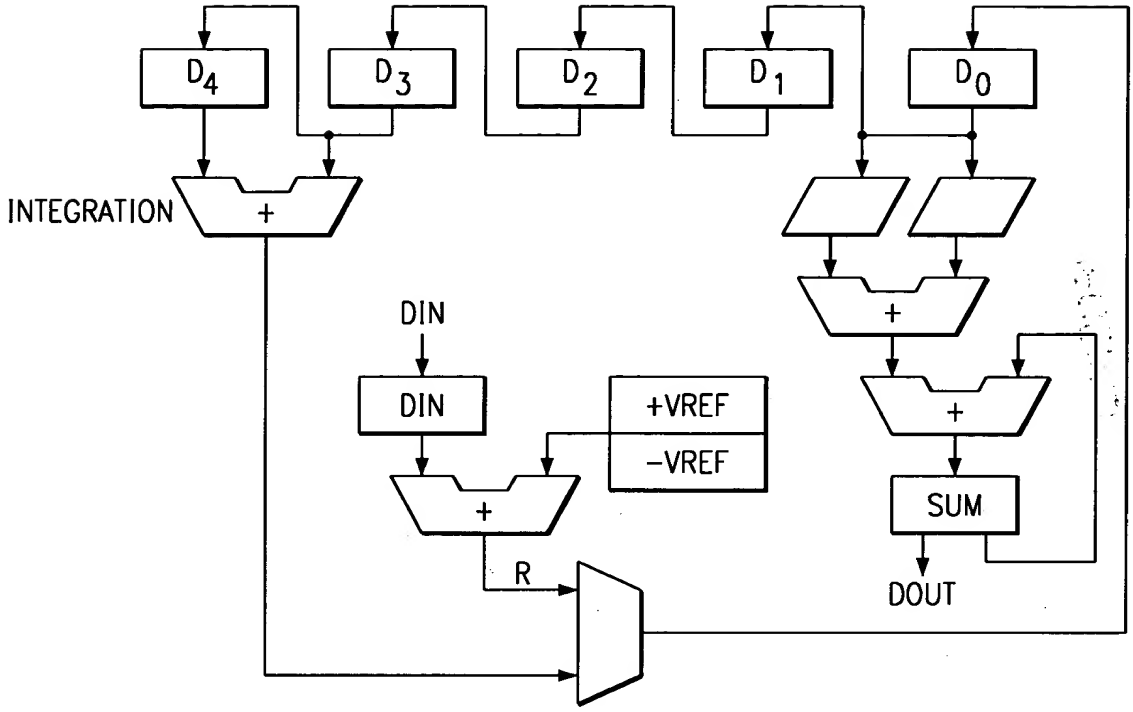


FIG. 30F-2

STATE	ACTIONS DURING STATE		
S0	$D_0(D4_k) = D_4(D4_{k-1})+D_3(D3_{k-1})$	CLEAR SUM	LOAD DIN_k
S1	$D_0(D3_k) = D_4(D3_{k-1})+D_3(D2_{k-1})$	SUM_k += $D_0(D4_k)$>>Shift4	
S2	$D_0(D2_k) = D_4(D2_{k-1})+D_3(D1_{k-1})$	SUM_k += $D_0(D3_k)$>>Shift3	
S3	$D_0(D1_k) = D_4(D1_{k-1})+D_3(R_{k-1})$	SUM_k += $D_0(D2_k)$>>Shift2	
S4		SUM_k += $D_0(D1_k)$>>Shift1	
S5	$D_0(R_k) = DIN_k$ +/- VREF		

Tohtsu-5f3e0860

FIG. 30G-1

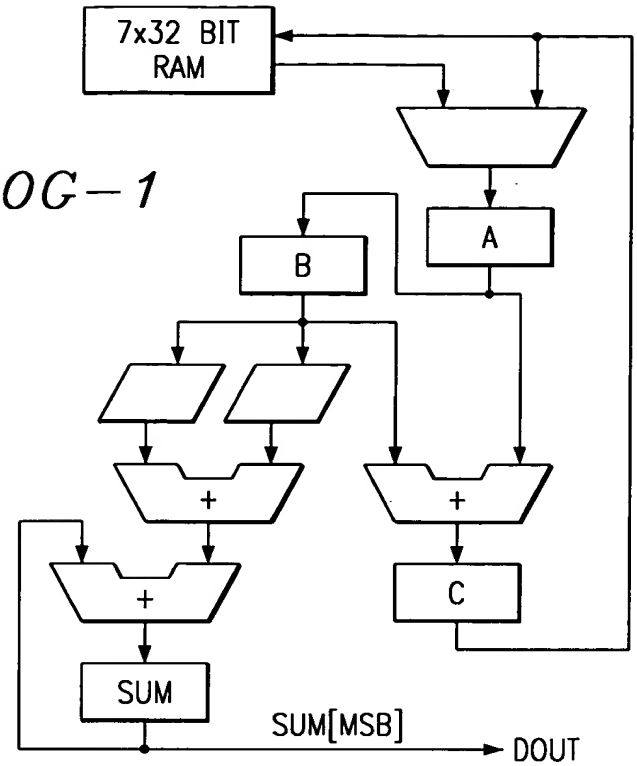
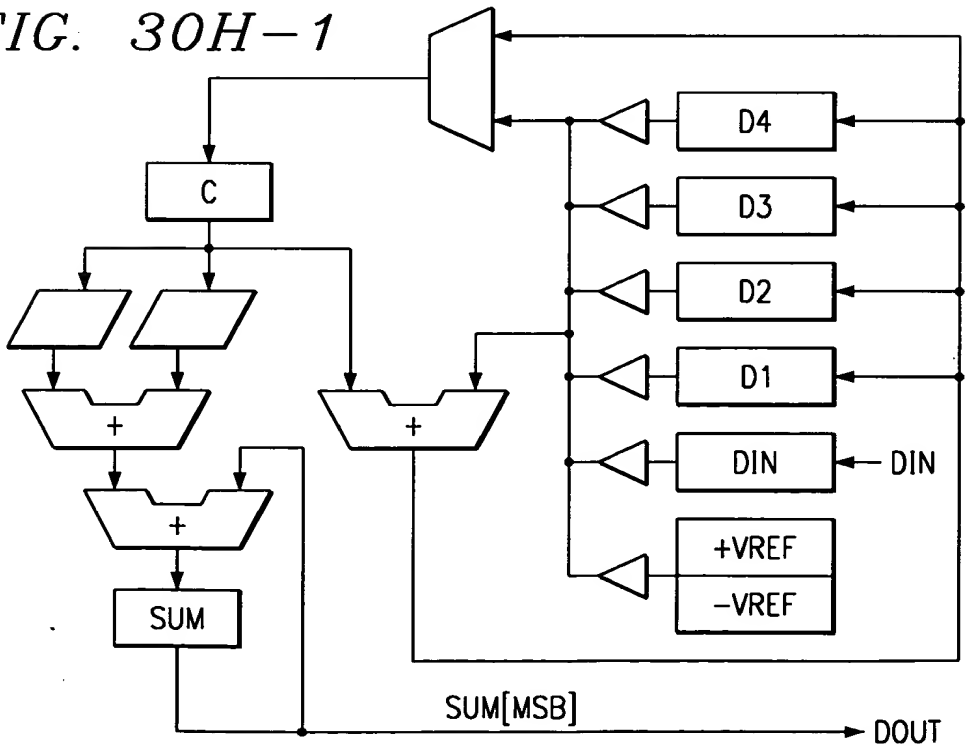


FIG. 30H-1



T04T90 "6th" 09/03/2004

STATE	ACTIONS DURING STATE			
S0	CLEAR SUM	CLEAR C	CLEAR B	CLEAR A
S1				LOAD A<Mem($D4_k$)
S2			SHIFT B<A($D4_k$)	LOAD A<Mem($D3_k$)
S3	SUM_k += B($D4_k$)>>Shift4	C = B($D4_k$) + A($D3_k$)	SHIFT B<A($D3_k$)	LOAD A<Mem($D2_k$)
S4				STORE C>Mem($D4_{k+1}$)
S5	SUM_k += B($D3_k$)>>Shift3	C = B($D3_k$) + A($D2_k$)	SHIFT B<A($D2_k$)	LOAD A<Mem($D1_k$)
S6				STORE C>Mem($D3_{k+1}$)
S7	SUM_k += B($D2_k$)>>Shift2	C = B($D2_k$) + A($D1_k$)	SHIFT B<A($D1_k$)	LOAD A<Mem(DIN_k)
S8				STORE C>Mem($D2_{k+1}$)
S9	SUM_k += B($D1_k$)>>Shift1	C = B($D1_k$) + A(DIN_k)	SHIFT B<A(DIN_k)	LOAD A<Mem(VREF)
S10			SHIFT B<A(VREF)	LOADREG A<C(TEMP)
S11		C = +/- B(VREF) + A(TEMP)		
S12				STORE C>Mem($D1_{k+1}$)

FIG. 30G-2

T0hT90" 6hEE0860

STATE	ACTIONS DURING STATE			
S0	CLEAR SUM	LOAD C < $D4_k$		LOAD DIN_k
S1	SUM_k += C($D4_k$)>>Shift4	LOAD C < $D3_k$	$D4_{k+1}$ = C($D4_k$) + $D3_k$	
S2	SUM_k += C($D3_k$)>>Shift3	LOAD C < $D2_k$	$D3_{k+1}$ = C($D3_k$) + $D2_k$	
S3	SUM_k += C($D2_k$)>>Shift2	LOAD C < $D1_k$	$D2_{k+1}$ = C($D2_k$) + $D1_k$	
S4	SUM_k += C($D1_k$)>>Shift1	C(TEMP) = C($D1_k$) + DIN_k		
S5			$D1_{k+1}$ = C(TEMP) +/- VREF	

FIG. 30H–2

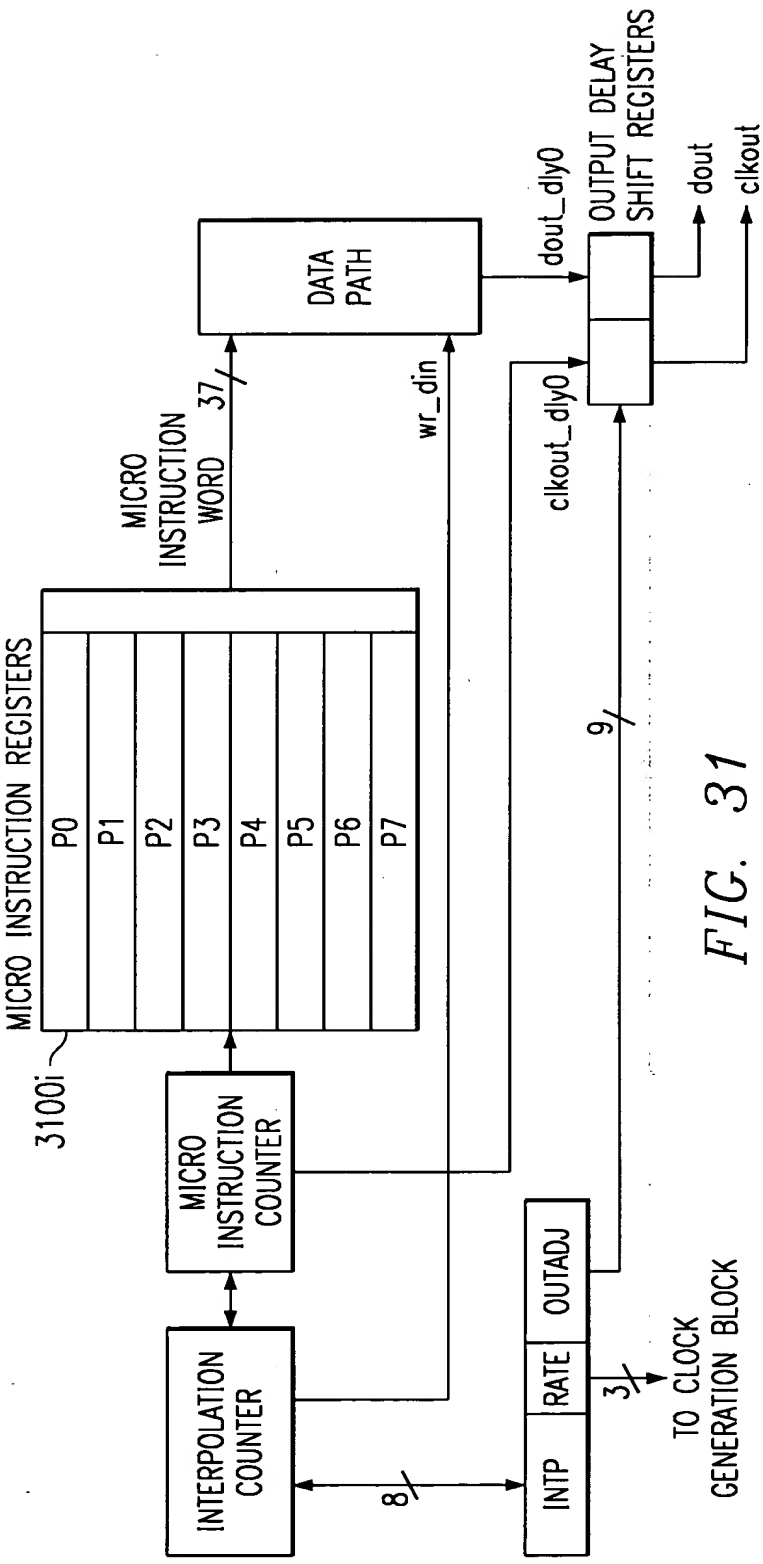

MICRO INSTRUCTION REGISTERS
P0
P1
P2
P3
P4
P5
P6
P7
3100i
MICRO INSTRUCTION WORD
37
DATA PATH
wr_din
dout_dly0
OUTPUT DELAY SHIFT REGISTERS
dout
clkout
clkout_dly0
9
MICRO INSTRUCTION COUNTER
INTERPOLATION COUNTER
8
INTP
RATE
OUTADJ
3
TO CLOCK GENERATION BLOCK

FIG. 31

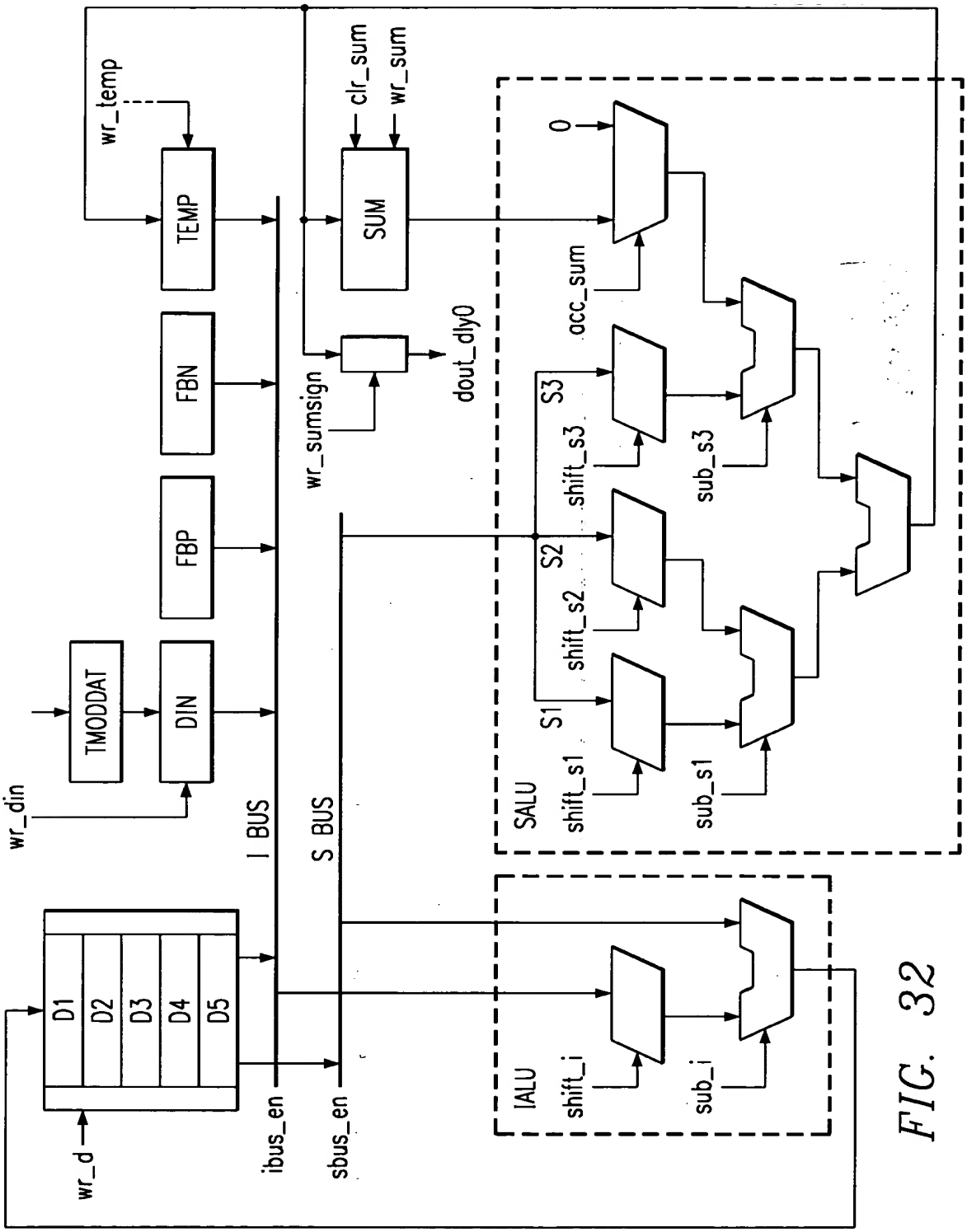

wr_temp
TEMP
FBN
FBP
TMODDAT
DIN
wr_din
I BUS
S BUS
wr_d
ibus_en
sbus_en
D1
D2
D3
D4
D5
clr_sum
wr_sum
SUM
wr_sumsign
dout_dly0
0
acc_sum
S3
shift_s3
sub_s3
S2
shift_s2
S1
shift_s1
sub_s1
SALU
IALU
shift_i
sub_i

FIG. 32

P	Feedforward	INTEGRATION	TEMP	DIN	SUM	SUMSIGN	TEMP	S BUS	I BUS	WRITE I
0	$SUM_k = D4_k>>11 + D4_k>>9 + D4_k>>7$	$D4_{k+1} = D4_k + D3_k$		LOAD DIN_k	WRITE			+D4>>7 +D4>>9 +D4>>11	+D3	D4
1	$SUM_k = SUM_k + D3_k>>8 + D3_k>>5 + D3_k>>4$	$D3_{k+1} = D3_k + D2_k$			ACC./ WRITE			+D3>>4 +D3>>5 +D3>>8	+D2	D3
2	$SUM_k = SUM_k + D2_k>>1 = D2_k>>7 = D2_k>>4$	$D2_{k+1} = D2_k + D1_k$			ACC./ WRITE			-D2>>4 +D2>>1 -D2>>7	+D1	D2
3	$SUM_k = SUM_k + D1_k$	$D1_{k+1'} = D1_k + DIN_k$			ACC./ WRITE	WRITE		+D1 +D1 -D1	+DIN	D1
4		$D1_{k+1} = D1_{k+1'}$ +/- VREF							+FB	D1
5										
6										
7										

FIG. 33

FIG. 34A

G	C	S	T	wr_d			fbus_en			sbus_en			shift_i					si	shift_s1					ss1	shift_s2					shift_s3					ss3	A	HEX
36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	1	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0	1	0	1	0	1	1	0	0	051C 00E4AC
0	0	1	0	0	1	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0	1	04D3 0082A1
0	0	1	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	048A 00909F
1	0	1	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1469 000003
0	0	0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0079 000000
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000 000000
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000 000000
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000 000000

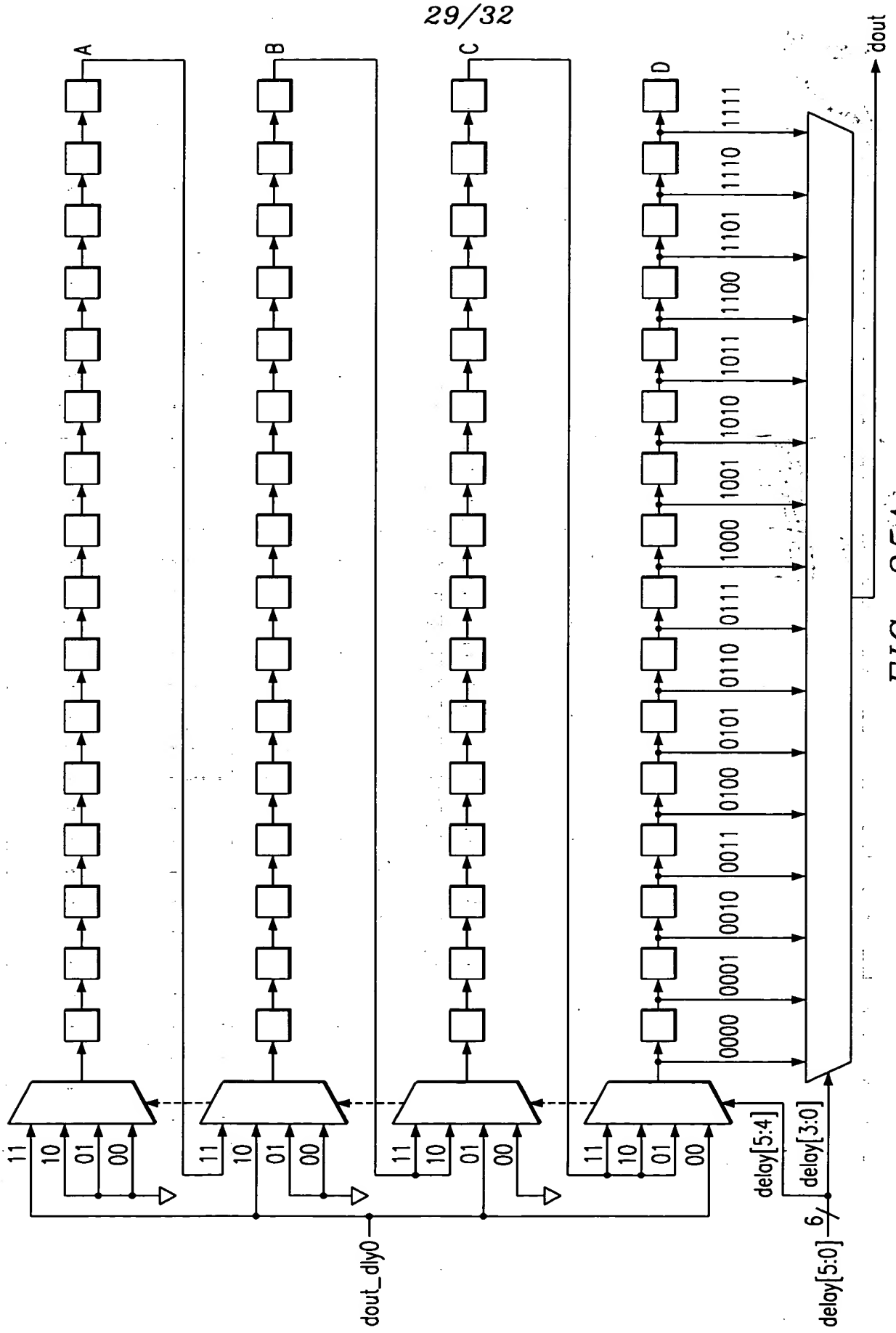

A
B
C
D
dout
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
11
10
01
00
dout_dly0
delay[5:4]
delay[3:0]
delay[5:0]
6

FIG. 35A

dout_dly0	DATA OUTPUT BIT, 0 DELAY
dout	DATA OUTPUT BIT, 0–63 CLOCK DELAY
delay[5:0]	HOW MANY CLOCKS (0–63) TO DELAY OUTPUT DATA dout_dly0
delay[5:4]	SELECTS SEGMENT INTO WHICH TO DIRECT dout_dly0
delay[3:0]	SELECTS WHERE TO TAP SEGMENT D TO GET dout

FIG. 35B

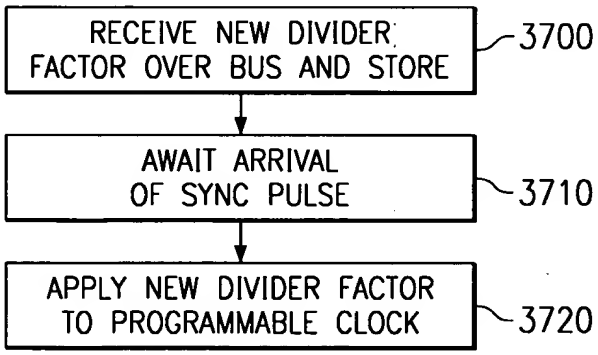


FIG. 37

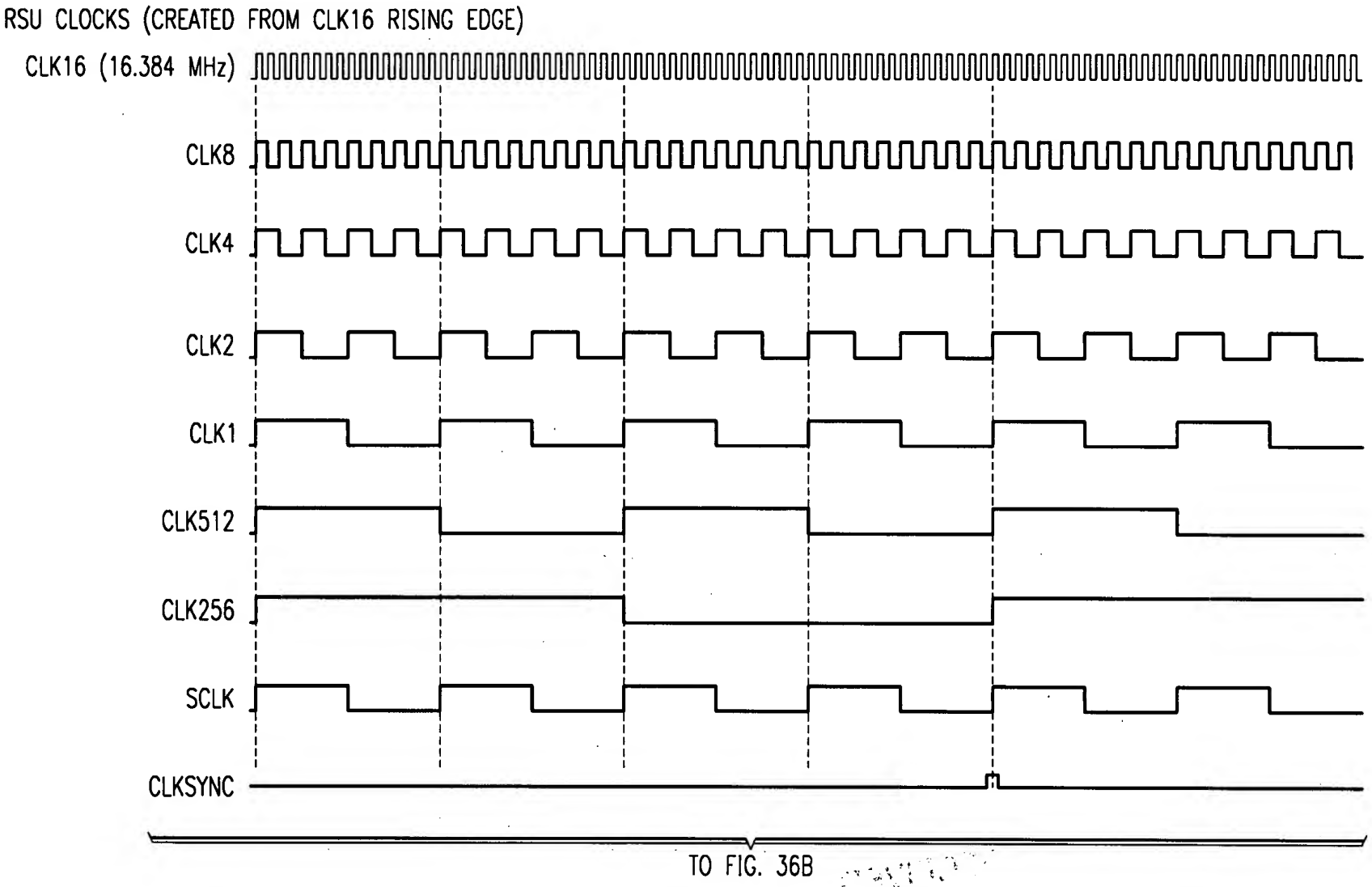

RSU CLOCKS (CREATED FROM CLK16 RISING EDGE)
CLK16 (16.384 MHz)
CLK8
CLK4
CLK2
CLK1
CLK512
CLK256
SCLK
CLKSYNC
TO FIG. 36B

FIG. 36A

T04T90 "6hE0860

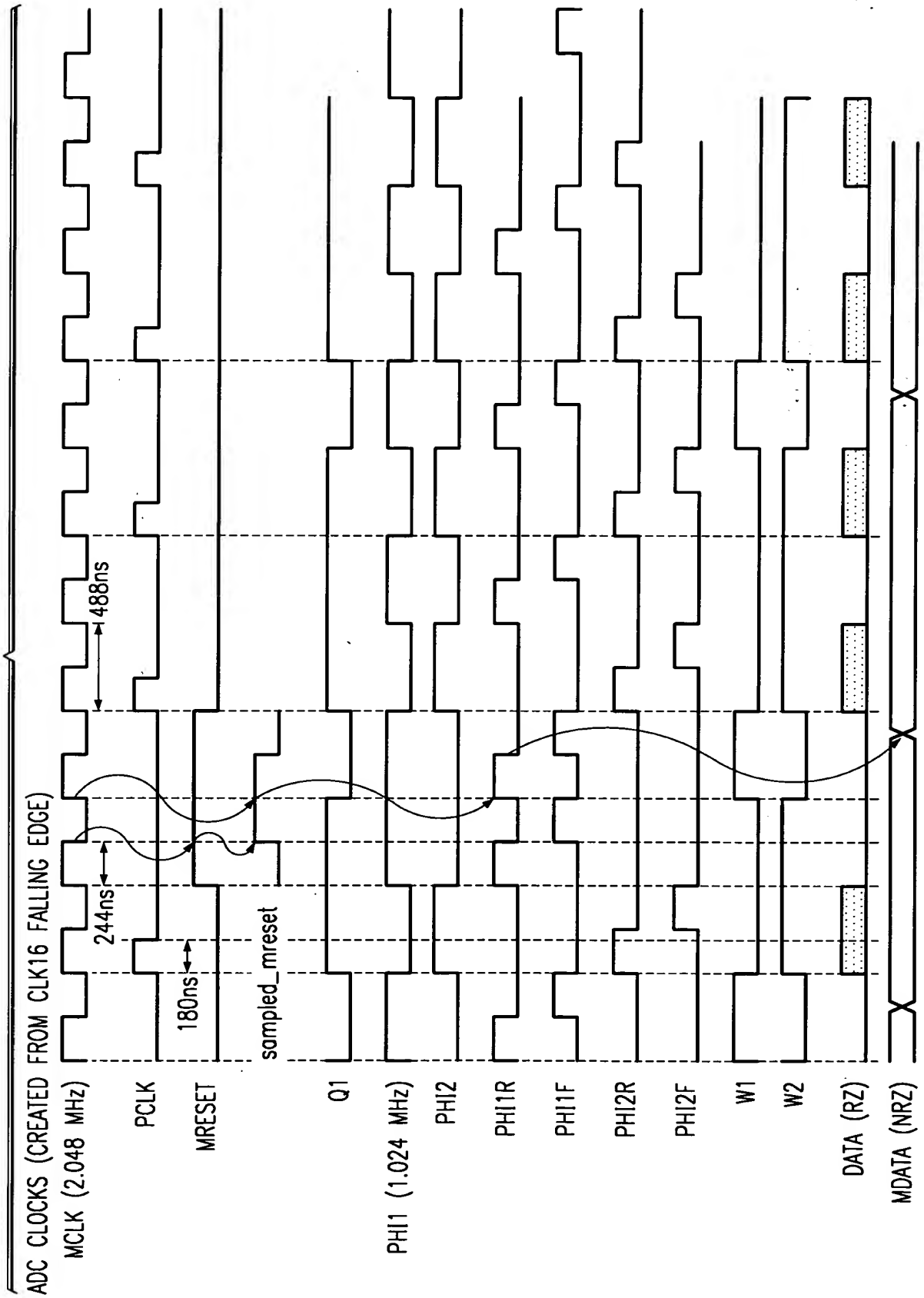



FIG. 36B